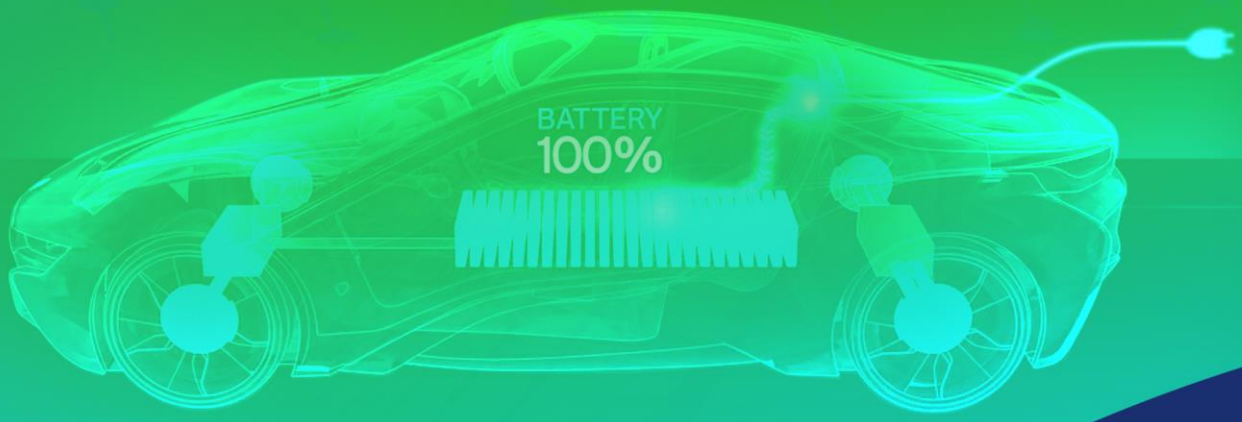
行云新能®
inwinic

新能源和智能网联汽车技术师资能力提升培训

主办单位：行云新能科技（深圳）有限公司
培训时间：03月15日-03月15日



行云桥微信公众号

培训实训手册

PRACTICAL TRAINING MANUAL

新能源和智能网联汽车技术
师资能力提升培训内容

培训日期	培训模块	培训内容
3月13日	智能化技术	行云大黄蜂智能化装备装调
		行云大黄蜂ADAS系统标定测试
		行云大黄蜂整车调测
3月14日		行云大黄蜂综合道路测试
3月15日	电动汽车技术	2020款比亚迪秦EV技术特征解析
		低压供电不正常故障诊断
		不能上“OK”电故障诊断
		不能正常行驶故障诊断
		交流不能充电故障诊断

紧急联系人:李洁/130 0663 7319

目录

一、智能网联汽车平台模块概述 1
（一）功能概况 1
二、智能网联汽车线控底盘及遥控器 2
（一）线控底盘 2
（二）WFLY 遥控说明 4
（三）使用与操作 5
（四）充电 6
三、智能网联汽车智能模块安装及线束连接 7
（一）智能模块配件清单 7
（二）车辆电气结构连接图 9
（三）安装毫米波雷达 10
（四）安装双目摄像头 11
（五）安装 RTK 天线 12
（六）安装激光雷达 13
（七）安装四个 360 全景摄像头 14
（八）安装平板电脑 16
（九）连接电源线、CAN 总线 19
（十）安装 ADAS 域控制器 20
（十一）安装超声波雷达 21
（十二）安装组合导航主机 21
（十三）安装计算单元 23
（十四）安装 4G 路由器 25
（十五）安装及线束连接完成 26
四、智能网联汽车软件平台参数设置及验证 28
（一）核心模块 28
（二）模块通信 29
（三）系统启动 30
（四）CAN 检测 31
（五）GPS 配置向导设置及验证 32
（六）CANBUS 测试 34
（七）激光雷达配置、检测 37
（八）毫米波雷达配置、检测 43
（九）IMU 配置、检测 44
（十）GPS 接收机配置、检测 45
（十一）超声波雷达配置、检测 46
（十二）定位模块配置、检测 47
（十三）双目摄像头检测及标定 48
（十）3D-AVM360 环视全景标定 60
五、智能网联汽车综合道路测试 67

（一）地图数据采集 67
（二）添加红绿灯 70
（三）自动驾驶功能测试 81

一、智能网联汽车平台模块概述

（一）功能概况

智能网联汽车硬件分为车辆上装和下装两部分（如图示 1.1）。

图 1.1　智能网联汽车硬件部分

上装主要有 16 线激光雷达、超声波雷达、毫米波雷达、摄像头（双目、环视）、GPS/惯导、自动驾驶处理器、ADAS 域处理器、CAN 卡、4G 路由器、装调台架、滑轨、调节机构和显示屏组成。

下装主要是车辆线控底盘。采用前轮阿克曼转向，后轮驱动结构；整体桥式悬挂加非承载式车身设计；在制动系统上采用电磁抱闸+液压碟刹结构；

在当前架构下，车辆可实现自动驾驶模式和人工接管模式。自动驾驶模式：可实现对障碍物识别分类，主动停障、避障；正确识别红绿灯，实现红灯停、绿灯行等动作（如图示 1.2）。人工接管模式：操作者通过遥控器，可控制车辆前进、后退、转向、紧急制动等。

图 1.2　智能网联汽车应用场景

二、智能网联汽车线控底盘、遥控器介绍

（一）线控底盘

线控底盘车结构（如图示 2.1），包括动力电池及电池管理系统、电机驱动与减速箱系统、电机控制器、线控转向电机、阿克曼转向机构、制动系统、电气系统及 CAN 网络通讯系统。

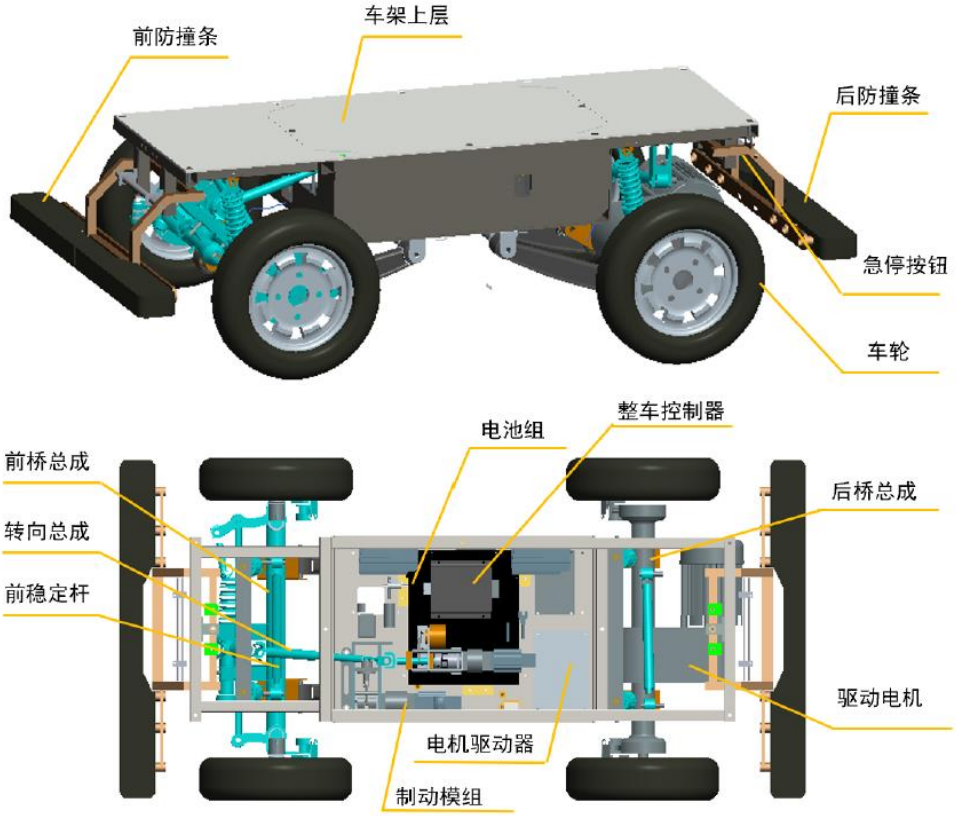


图 2.1　线控底盘车结构

本智能网联汽车选用底盘可以通过电压电量显示器及启动声音来确定车体的启动状态。

表 2.1　车体状况描述

状态	描述
当前电压	当前电池电压可通过尾部显示器电压电量表查看
上电显示	按下电源开关，出现“嗒”的一声，并且尾部显示屏全部亮起

底盘顶部的电气位置（如图示 2.2），从底盘引出两个接口分别是上装总供电电源、底盘 CAN 接口，（如图示 2.3）。此智能网联汽车 CAN 功能选择引脚编号为 1CAN 总线高，3CAN 总线低；电源选择引脚编号为 A3 电压正，电压 12V，A4 电源负（其它引脚为预留口，暂不使用）。

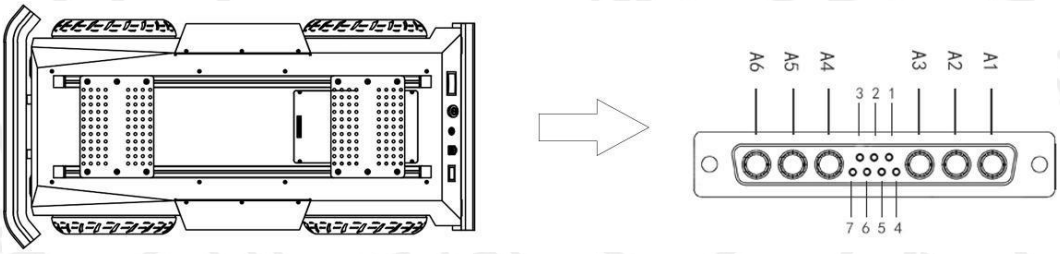


图 2.2　顶部电气位置示意图

引脚编号	引脚类型	功能及定义	备注
1	CAN	CAN_H	CAN总线高
3		CAN_L	CAN总线低
A1	电源	VCC	电源正,电压5V
A2		GND	电源负
A3		VCC	电源正,电压12V
A4		GND	电源负
A5		VCC	电源正,电压24V
A6		GND	电源负
注：24567引脚处于预留状态，未启用			

图 2.3　顶部电气接口具体引脚定义

尾部的电气板（如图示 2.4），其中 B1 为电压显示器（电量），实时显示电池电压；B2 为充电接口；B3 为电源总开关；B4 为急停旋钮。

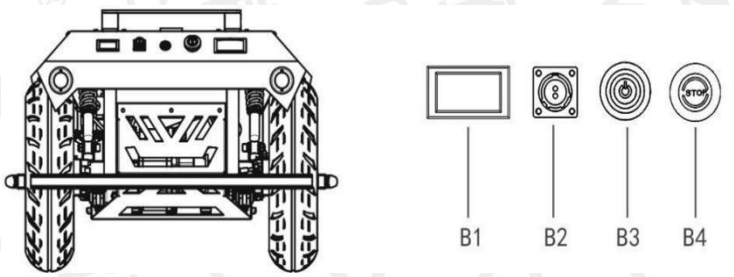


图 2.4 尾部视图及电气面板

（二）WFLY 遥控功能说明

遥控器在产品出厂前均已经对码成功，无需改动设置。

1. WFLY 遥控控制说明

每台底盘配备一个 WFLY ET07 遥控器，用户使用遥控器可以控制底盘运动，在本产品中的 WFLY ET07 遥控器我们采用左手控制制动，右手控制前、后油门，左、右转向的设计。其定义及其功能（如图示 2.5）。

图 2.5　WFLYET07 遥控器主要按键示意图

按键的功能定义为：SC、SD、V1、T4 暂时未被启用；参数设置在出厂前均已经设置好，禁止私自修改系统设置。

其中智能网联汽车主要使用按键功能如下：

（1）SA 为控制模式切换拨杆，两档位，拨杆向下时为遥控器控制模式，拨杆向上时切换为自动控制模式；

（2）S1 为制动摇杆，上推刹车，下推到底为松开刹车；

（3）S2 前后为油门控制，控制底盘前进和后退；S2 左右控制前轮的转向；

（4）SB 为档位切换拨杆，共三个档位。拨杆在中位时，车体不接收前、后运动控制信号；拨杆往上拨时，切换至前进挡，底盘才能接收 S2 油门摇杆发送的前进运动控制信号，往前运动；拨杆往下拨、切换至后退档，底盘才能接收 S2 油门摇杆发送的后退运动控制信号，往后运动；

（5）V2 为驻车请求拨盘，拨盘往下拨发送驻车请求，启动驻车制动装置；拨盘往上拨，即松开驻车制动装置。

（6）POWER 为电源按钮，长按进行开、关机操作。

2.遥控器蜂鸣器警告说明（如表 2.3）

表 2.3 遥控器报警状态说明

低电压报警	电量低于默认设置电压时，蜂鸣器响，遥控器显示屏右上角电压值闪烁，并显示警告界面，按“是”可正常使用，报警声长鸣，按“否”则关机。
开关位置报警	开机时，若油门 SA/SB/SC 拨杆不是在默认挡位时（默认，将会出现在什么位置）报警界面（显示对应开关）， 所有开关在复位则界面消失。
关机报警界面	遥控器关机时，会检测接收器是否通讯，通讯则弹出警告界面并且需要经过确认才能关机。
微调提示音	微调键微调值达到中点或端点时，蜂鸣器长响一声。

（三）遥控器使用操作说明

1.检查

检查车体状态。检查车体是否有明显异常；

检查急停开关状态，确认尾部急停按钮均处于释放状态；

检查遥控器的所有档位均处于空挡位置。

2. 启动

底盘车体与建立的参考坐标系 X 轴为平行状态，如图 3.6 示。

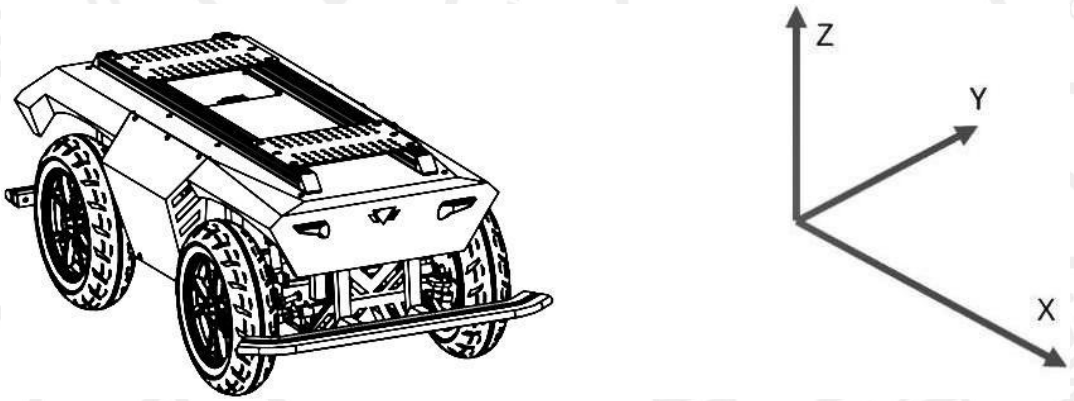


图 3.6 标准建立坐标参考系

按下 B3（电源总开关按钮），确保遥控器急停开关都关闭（S1 摇杆拉到最下方，V2 拨盘向下拨动，使车辆尾灯不亮情况下），将 SB 拨杆拨向前，（遥控器摇杆 S2 往前推动）则车辆往 X 正方向运动；将 SB 拨杆拨向后（S2 往后推动），则车辆往 X 负方向运动；SB 拨杆拨动在中间时，车辆处于空挡状态不接收前后运动控制信号。车辆速度大小和 S2 推动程度成正比。遥控器摇杆 S2 左右控制车体前轮的转向运动，S2 往左推，小车往左转向，推至最大，此时左转向速度最大；S2 往右推，小车往右转，推至最大，此时右转向速度最大。

注：检查电池电压，查看电池电压是否正常，若电压低于 20%，请先充电；

3.关闭操作

再次长按 B3（电源总开关按钮），释放开关，即可关闭遥控器电源。

（四）充电

汽车底盘默认配备一个充电器，可满足用户的充电需求。

充电具体操作流程如下：

1.充电前请确保底盘处于停机、断电状态，确认尾电气板中 B3（电源总开关）处于关闭状态；

2.先将充电器的输出插头插入尾部电气板中 B2 充电接口中；再将充电器的 AC 插头插入 220V 交流电插座。充电结束之后反之，先拔 AC 插头，再拔输出端插头。

图 2.7　车辆充电器充电状态

下表 2.4、2.5 为充电器基本说明。

表 2.4 充电器状态 LED 指示灯说明

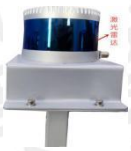
LED 指示灯状态	充电器状态
LED1 亮红色	充电器输入线接头已经通电
LED2 亮红色	表示充电器正在充电
LED2 亮绿色	表示电池已充满电

三、智能网联汽车智能模块安装、线束连接介绍

（一）智能模块配件清单

表 3.1　智能模块配件清单

配件清单	配件清单列表	器件数量	备注
计算单元	计算单元	1	

组合导航主机	组合导航主机	1	
惯导模块	惯导模块	1	
双目相机	双目相机	1	
超声波雷达	超声波雷达控制器	1	
激光雷达	16 线激光雷达	1	
毫米波雷达	毫米波雷达	1	
4G 路由器	4G 路由器	1	

SIM 卡	SIM 卡	1	
散热风扇	散热风扇	1	
RTK 天线	RTK 天线	2	
ADAS 域控制器	ADAS 域控制器	1	
平板电脑	平板电脑	1	
电源线控制板	电源线控制板	1	

（二）车辆电气结构连接图

车辆上装和下装电气连接图（如图示 3.1）。底盘分出电源总线和 CAN 线，电源总线连接电源控制板为整个车辆上装供电；CAN 线将处理器处理的结果传向底盘 VCU。电源控制板与毫米波雷达、4G 路由器、电源总开关、组合导航主机、激光雷达、超声波雷达、双目摄像头、ADAS 域控制器、平板电脑、显示器、计算单元、风扇相连，为上述零部件提供电源；底盘、毫米波雷达、超声波雷达和 ADAS 域控制器的 CAN 线与 CAN 收发器相连，CAN 收发器将收到的 CAN 信号发送给处理器进行处理，得到的处理结果通过 CAN 收发器传向底盘 VCU，

控制执行系统。感知模块的数据信息传递到自动驾驶处理器，处理后的结果，通过 CAN 收发器传送到底盘 VCU，进而控制车辆底盘运动。4G 路由器（外接 2 个天线）通过网线与组合导航主机、激光雷达、双目摄像头、自动驾驶处理器相连，保证各组件通讯正常；ADAS 域控制器与 360 高清全景摄像头相连，并与平板电脑相连。

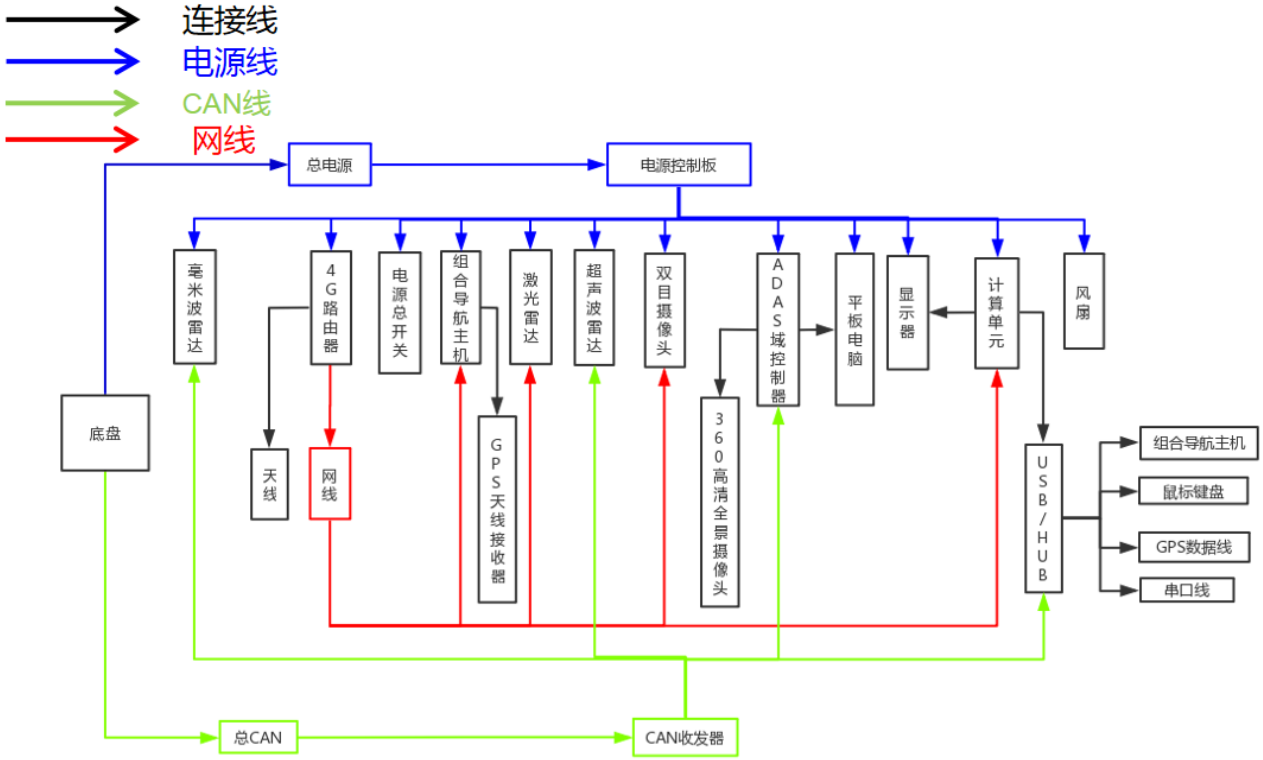


图 3.1　车辆整体线路结构连接图

注：1.组合导航主机与 GPS 天线接收器通过数据线相连。

2.计算单元和显示器相连，并外接 USB/HUB 数据插头。

（三）安装毫米波雷达

毫米波雷达安装位于车辆车头中间（如图示 3.2）。

1.工具选用 5 号内六角扳手，固定毫米波雷达的三颗固定螺栓。

图 3.2　毫米波雷达安装

2.连接毫米波雷达线束（如图示 3.3）。

图 3.3　毫米波雷达线束

（四）安装双目摄像头

双目摄像头位于智能车辆车头正中位置（如图示 3.5）。

1.工具选用 2.5 号内六角扳手，安装四颗固定螺栓固定在双目摄像头架上（如图示 3.4）。

图 3.4　双目摄像头

2.连接双目摄像头网线及电源线。

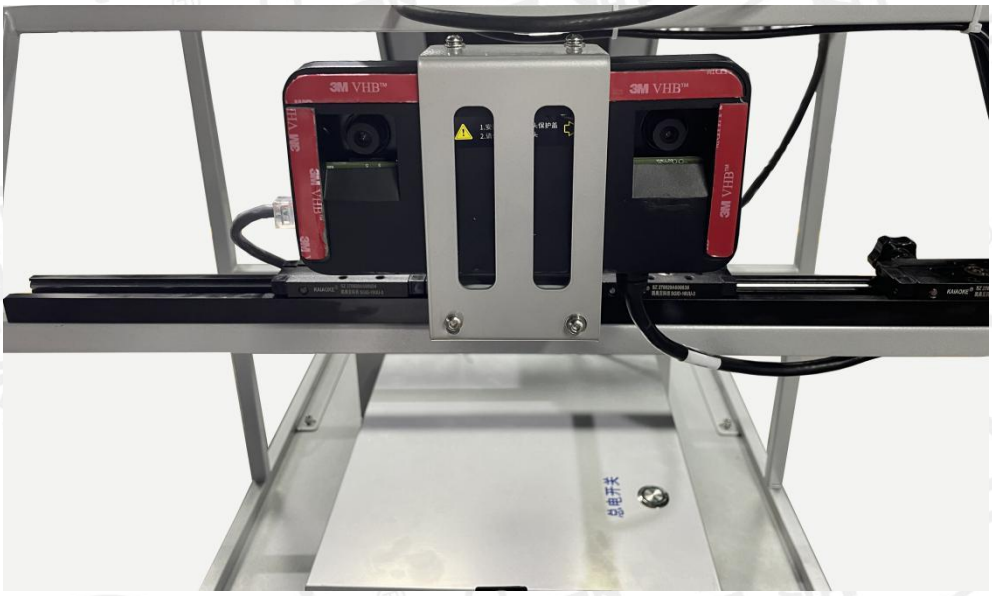


图 3.5　安装双目摄像头及网线

（五）安装 RTK 天线

RTK 天线接收器位于车体上装，车头车尾各一个。

1.工具选用 3 号、4 号内六角扳手，安装三颗固定螺栓在车体上装（如图示 3.6）。

图 3.6　安装 RTK 天线接收器

2.RTK 天线接口与 RTK 天线接头相连，RTK 控制模块与组合导航系统相连（如图示 3.7、3.8）。

图 3.7　RTK 天线接口

图 3.8　连接 RTK 天线

注意事项：

①安装 RTK 天线时注意选取合理工具，切不可失手掉落天线。

②RTK 天线接收器分前后，注意连接位置。

（六）安装激光雷达

激光雷达位于车体上装中间，并和激光雷达底座座固定（如图示 3.9、3.10.3.11）。

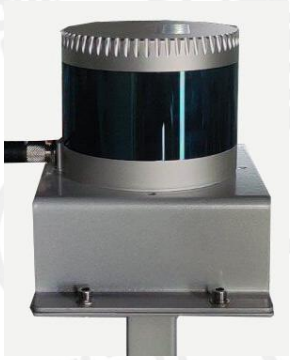


图 3.9　激光雷达与底座固定

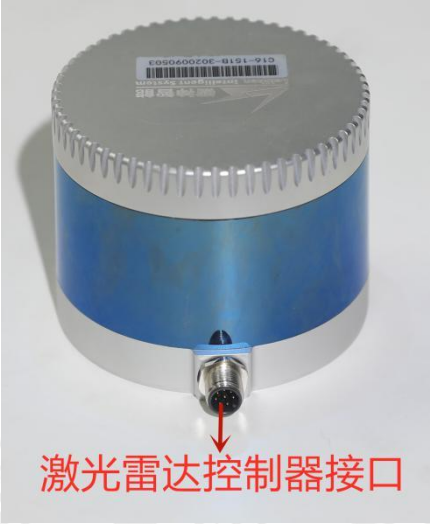


图 3.10　激光雷达控制器接口

图 3.11　激光雷达控制器

（七）安装四个 360 全景摄像头

360 高清全景摄像头位置分别安装于车体上装前、后、左、右中间位置（如图示 3.12、3.13、3.14、3.15）。

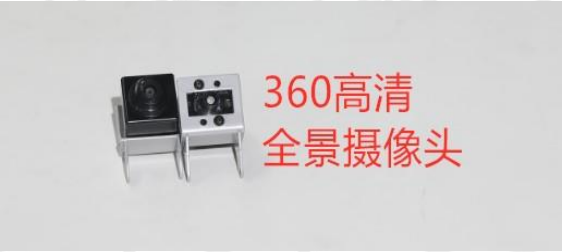
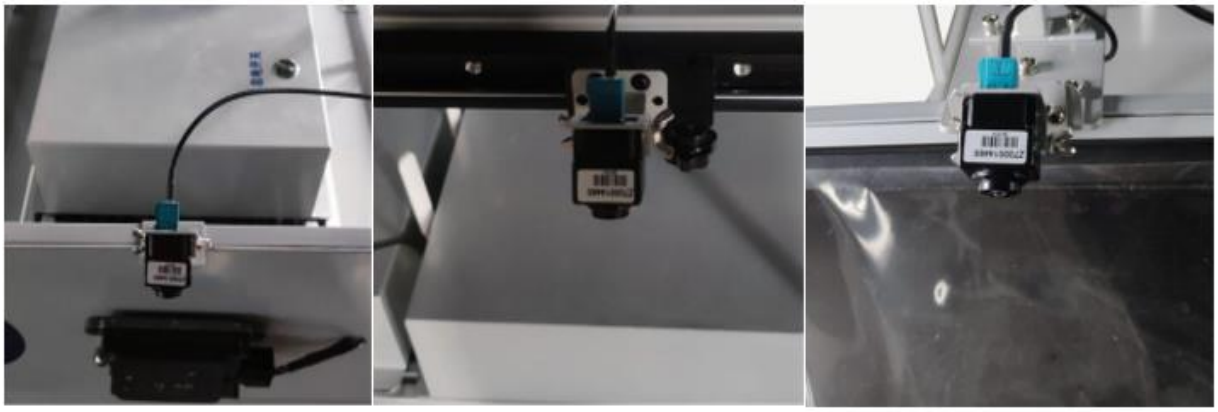


图 3.12　360 高清全景摄像头

1.把摄像头固定在摄像头架上，固定在车辆上装。



前摄像头　　左、右摄像头　　后摄像头

图 3.13　360 高清全景摄像头安装位置

2.360 高清全景摄像头接头和 ADAS 域控制器接口相连。

图 3.14　360 高清全景摄像头接头

图 3.15　360 高清全景摄像头接口位置

注意事项：

①注意摄像头四个摄像头方向，切勿装反。

（八）安装平板电脑

平板电脑安装在车体上方（如图示 3.16、3.17、3.18、3.19、3.20）。

工具选用 3 号内六角扳手，将平板电脑固定在平板电脑支架上，并固定在车体上装。

图 3.16　固定平板电脑

图 3.17　平板电脑接口

图 3.18　高清线接口

图 3.19　平板电脑通讯线和电源线

图 3.20　ADAS 域控制器通讯线接口和平板电脑高清线接口

注意事项：

①拔接端子注意不要暴力插拔，插拔前要看好端子内插针有无损坏、变形。

②安装平板电脑时切勿失手掉落。

（九）连接电源线、底盘 CAN 线

电源线、底盘 CAN 线接口位于底盘上（如图示 3.21、3.22、3.23、3.24）。

图 3.21　电源线、CAN 线接口

1.连接电源线、CAN 线接口。

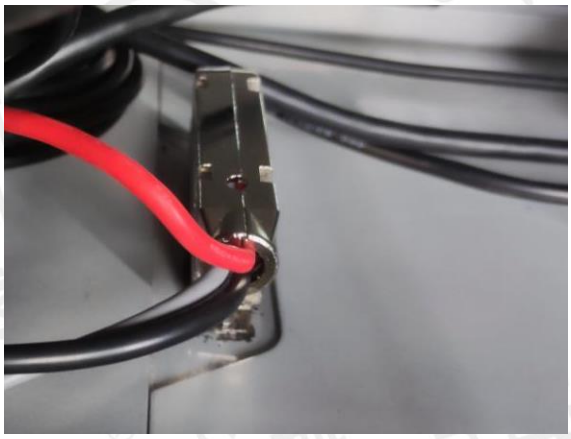


图 3.22　连接电源线、CAN 线

2.电源线、CAN 线和电路板相连。

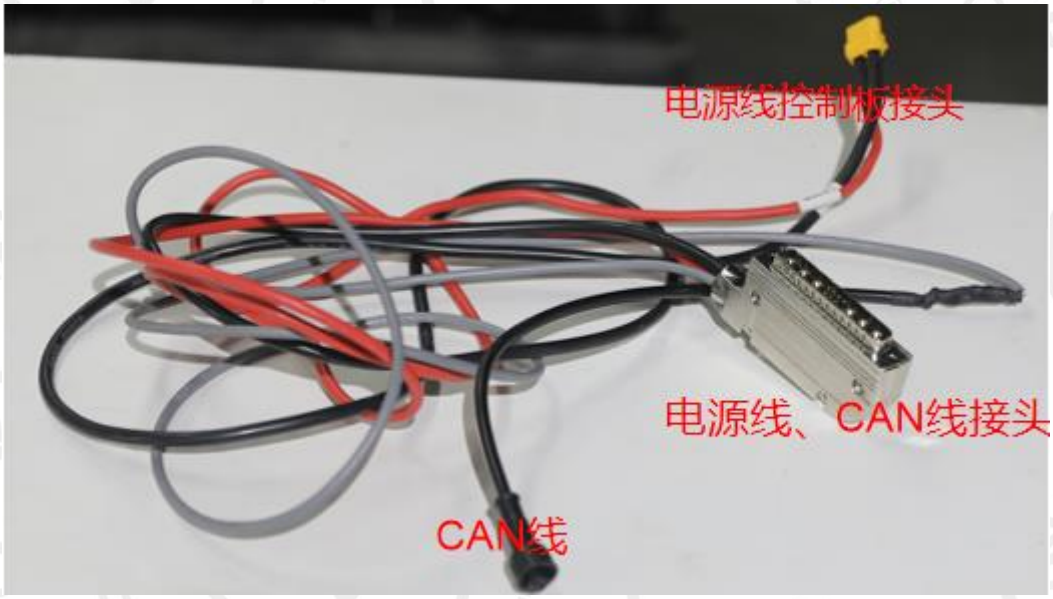


图 3.23　CAN 线总线

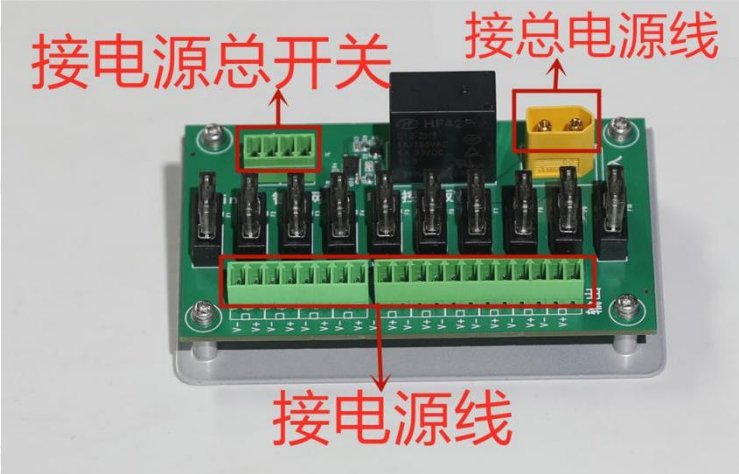


图 3.24　总电源电路板

（十）安装 ADAS 域控制器

ADAS 域控制器安装在智能网联汽车固定板上（如图示 3.25）。

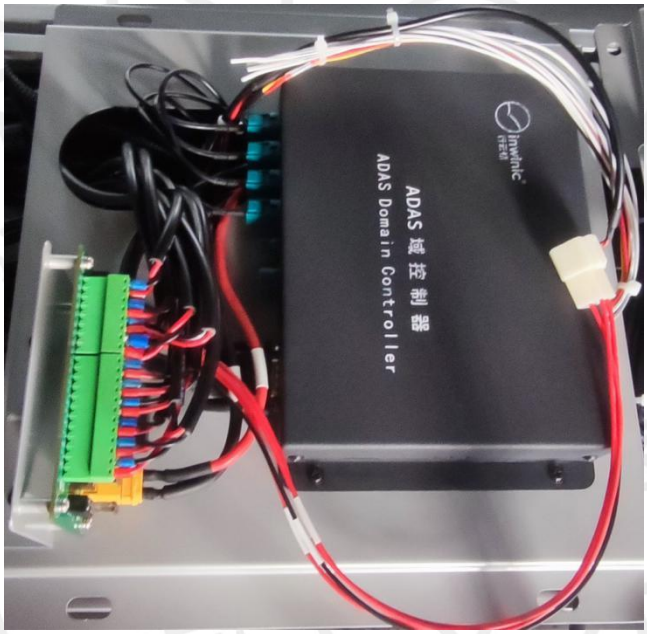


图 3.25　ADAS 域控制器安装

（十一）安装超声波雷达

超声波雷达控制器安装在智能网联汽车固定板上（如图示 3.26、3.27）。

图 3.26　超声波雷达控制器接口及固定空位

①电源线及 CAN 线连接孔位

②前左超声波雷达

③后右超声波雷达

图 3.27 超声波雷达连接线

注意事项：

①拔接端子注意不要暴力插拔，插拔前要看好端子内插针有无损坏、变形。

②插接超声波雷达插头时，注意两种插头上线位不同，插接进模块位置也不相同，注意不可暴力对插，以免造成模块内部端子插针变形。

（十二）安装组合导航主机

组合导航主机安装在智能网联汽车固定板上（如图示 3.28、3.29、3.30）。

图 3.28　组合导航主机

①惯导模块连接口

②接 RTK 前天线

③接 RTK 后天线

④一拖 4 连接孔

图 3.29　惯性导航模块接线

图 3.30　组合导航主机一托五

①组合导航主机接头

②电源线

③网线接口

④串口线接口

注意事项：

①拔接端子注意不要暴力插拔，插拔前要看好端子内插针有无损坏、变形。

②组合导航主机上表面的 Y 轴方向需要指向车辆前进方向

③注意拆装组合导航主机模块时避免工具掉落撞击该模块，取放模块时做到轻拿轻放，连接该模块线束时注意两端孔位内小螺栓不要过力旋钮，以免造成损坏。

④拆装 GPS 天线时注意选取合理工具，切不可失手掉落天线。

（十三）安装计算单元

计算单元安装在智能网联汽车固定板上（如图示 3.31、3.32、3.33、3.34、3.35、3.36）。

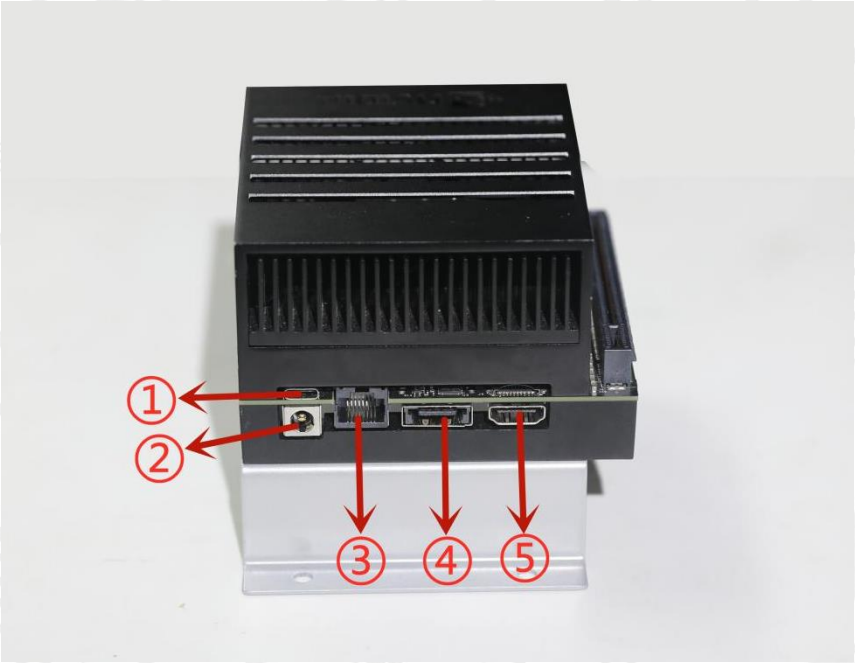


图 3.31　计算单元

①typc 转接口

②电源线接口

③网线接口

④USB 接口

⑤HDMI 高清连接线接口

图 3.32　高清连接线

图 3.33 主机电源线

图 3.34　USB/HUB

USB 集线器上接：鼠标接收器、CAN 收发器、串口线、惯导模块、GPS 数据线。

图 3.35　CAN 收发器

①USB 接口

②CAN 总线接口

图 3.36　CAN 总线

注意事项：

①拔接端子注意不要暴力插拔，插拔前要看好端子内插针有无损坏、变形。

②拆装计算单元注意轻拿轻放及注意安装方向。

③安装计算单元接线时注意正确连接限位，严禁暴力插接

（十四）安装 4G 路由器

4G 路由器安装在智能网联汽车固定板上（如图示 3.37、3.38）。

图 3.37　4G 路由器

①天线接口

②供电电源接口

③计算单元网线接口

④激光雷达控制器网线接口

⑤双目摄像头网线接口

⑥一拖 5 网线接口

图 3.38　路由器外接线

注意事项：

①拔接端子注意不要暴力插拔，插拔前要看好端子内插针有无损坏、变形。

②拆装路由器注意轻拿轻放及注意安装方向。

③安装路由器连接线时注意下方魔术贴粘接牢靠，严禁暴力插接

（十五）安装及线束连接完成

智能网联汽车固定板安装完成（如图示 3.39、3.40）。

图 3.39　智能网联汽车固定板

整理好线束，盖上盖板，车辆整体。

图 3.40　智能网联汽车安装完成

四、智能网联汽车软件平台参数设置及验证

（一）核心模块

在智能网联汽车实训平台上，运行的核心软件模块包括：

perception：感知模块，感知车辆周围的环境信息。包括障碍物检测、车道线检测、交通信号灯检测等。

prediction：预测模块用来预测感知障碍物的运动轨迹。

routing：routing 模块车辆如何通过一系列车道或道路到达目的地。

planning：planning 模块规划车辆的运动轨迹。

control：control 模块通过生成诸如节流阀，制动器和转向的控制命令来执行已规划的轨迹。

canbus：通过 canbus 模块将控制命令传递给车辆 VCU，控制底盘运动，同时也可将底盘信息传递给自动驾驶系统。

localization：定位模块利用各种感知和定位信息，确定车辆的位置。例如 GPS，LiDAR 和 IMU 提供的信息，来估计车辆所在的位置。

这些模块的交互结构（如图示 4.1）

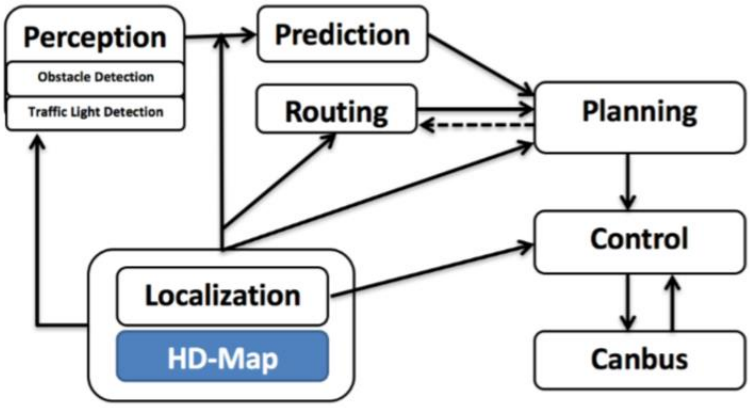


图 4.1 智能网联汽车核心模块的交互结构

每个模块都作为独立的 ROS 节点运行。每个模块节点都会发布和订阅某些主题。订阅的主题用作数据输入，而发布的主题用作数据输出。

（二）模块通信

智能网联汽车实训平台使用 Apollo 平台，Apollo 项目基于 ROS（Robot Operating System），ROS 包含了一套开源的软件库和工具，专门用来构建机器人应用。在一个 ROS 系统中，包含了一系列的独立节点（nodes），这些节点之间，通过发布/订阅的消息模型进行通信；ROS 系统中的节点可能位于不同的主机上；ROS 系统中包含了一个主（Master）节点，主节点使得其他节点可以查询彼此以进行通讯，所有节点都需要在主节点上进行注册，然后就可以与其他节点通讯了。Apollo 对 ROS 进行了改造，主要包括下面三个方面：1.通信性能优化；2.去中心化网络拓扑；3.数据兼容性扩展。

1. 通信性能优化

自动驾驶车辆中包含了大量的传感器，这些传感器可能以非常高频的速度产生数据，所以整个系统对于数据传输效率要求很高。在 ROS 系统中，从数据的发布到订阅节点之间需要进行数据的拷贝。在数据量很大的情况下，很显然这会影响数据的传输效率。所以 Apollo 项目对于 ROS 第一个改造就是将通过共享内存来减少数据拷贝，以提升通信性能（如图示 4.2）。

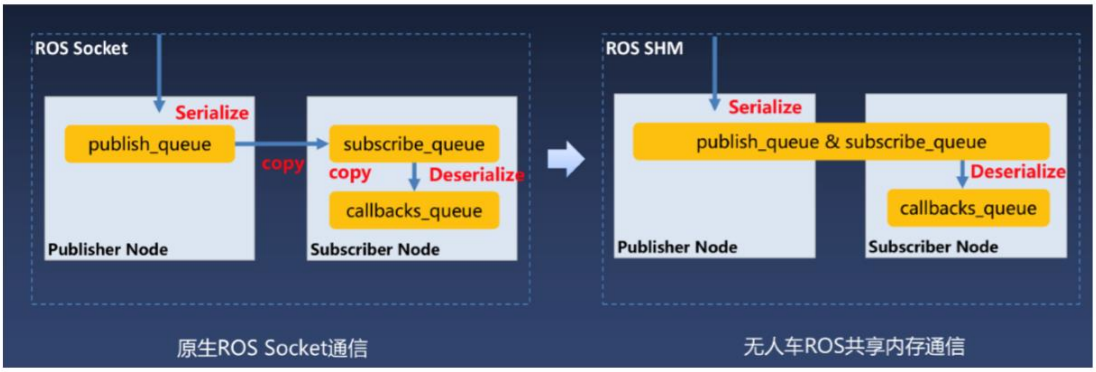


图 4.2　智能网联汽车通讯

2.去中心化网络拓扑

ROS 系统中包含了一个通信的主节点，所有其他节点都要借助于这个节点来进行通信。所以，很显然的，假如这个节点发生了通信故障，就会影响整个系统的通信。并且，整个结构还缺乏异常恢复机制。所以 Apollo 项目对于 ROS 的第二个改造就是去除这种中心化的网络结构。Apollo 使用 RTPS（Real-Time Publish-Subscribe）服务发现协议实现完全的 P2P 网络拓扑。

3.数据兼容性扩展

Apollo 项目对于 ROS 最后一个较大的改进就是对于数据格式的调整。在 ROS 系统中，使用 msg 描述文件定义模块间的消息接口。但不幸的是，接口升级之后不同的版本的模块难以兼容。因此，Apollo 选择了 Google 的 Protocol Buffers 格式数据来解决这个问题。Protocol Buffers，是 Google 公司开发的一种数据描述语言，类似于 XML 能够将结构化数据序列化，可用于数据存储、通信协议等方面。它不依赖于语言和平台并且可扩展性极强。现阶段官方支持 C++、JAVA、Python 三种编程语言，但可以找到大量的几乎涵盖所有语言的第三方拓展包。

（三）系统启动

智能网联汽车实训平台中已经预装开发和编译环境，是以 Docker 镜像的形式部署在计算单元中。计算单元默认配置如下（如图示 4.3）：

IP 地址	远程登录方式	用户名/密码	Windows 远程登录工具	Linux/Mac 远程登录工具
192.168.1.9/24	SSH/VNC	nvidia/nvidia	Putty、XShell、MobaXterm、SecureCRT	终端、SecureCRT

图 4.3 计算单元默认配置

1.打开终端，切换到 Apollo 源码所在目录，执行 dev_into.sh 脚本进入 docker 容器（如图示 4.4）。

$ cd work/ApolloAuto/apollo

$ ./docker/scripts/dev_into.sh

```
nvidia@tegra-ubuntu:~/work/ApolloAuto/apollo$ ./docker/scripts/dev_into.sh
nvidia@in_dev_docker:/apollo$
```

图 4.4 进入 docker 指令

2.完成后执行 bootstrap.sh 脚本启动 Apollo。

$ ./scripts/bootstrap.sh

3.在浏览器中访问 localhost:8888 或 192.168.1.9:8888（192.168.1.9 是计算单元的 IP 地址）；看到 Apollo 前端界面 Dreamview 表示正常启动（如图示 4.5）。

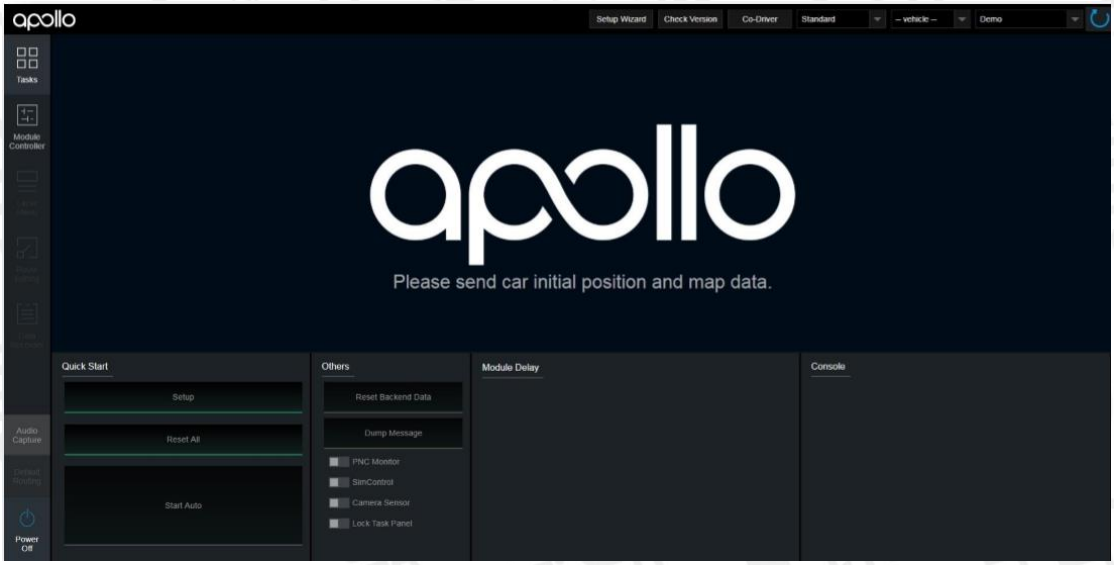


图 4.5　进入 Apollo 前端界面 Dreamivew

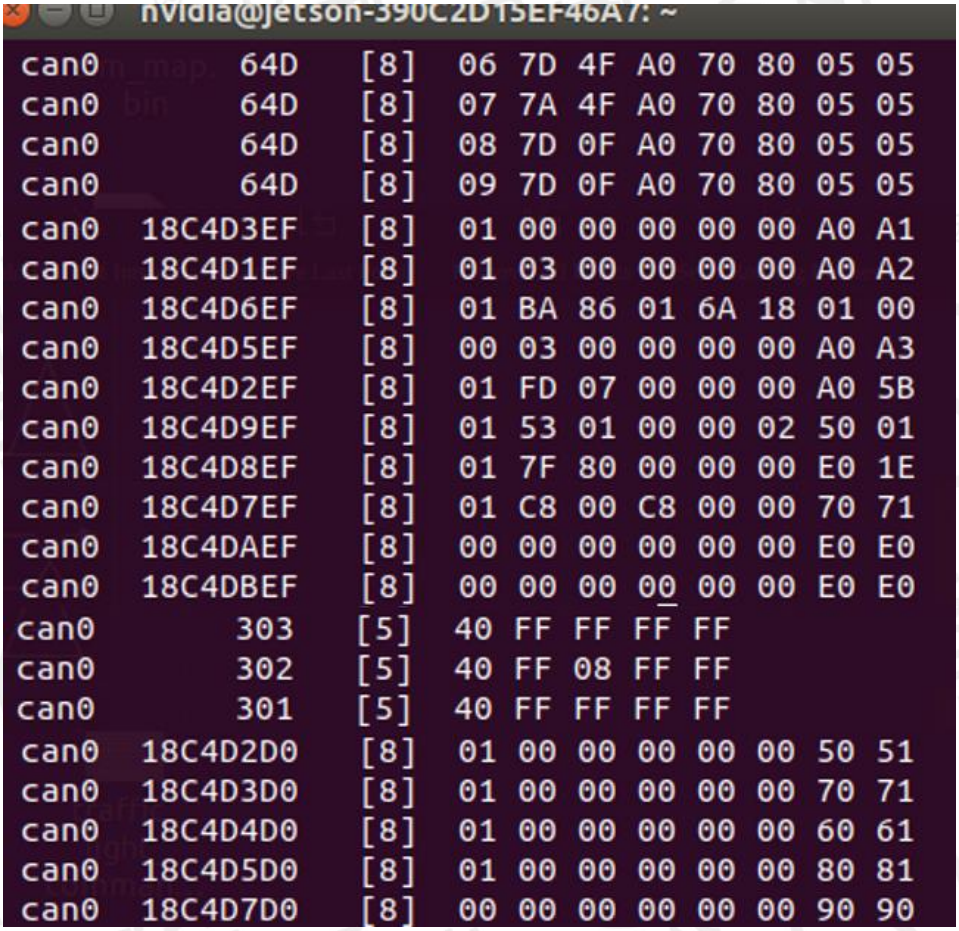
（四）CAN 检测

CAN 收发器，它将底盘提供的数据转化为电信号并通过数据总线发送出去，同时也接收总线数据。

将线束连接完成后，通过 CAN 检测监听，检查 CAN 卡与 ADAS 域控制器、底盘 CAN、毫米波雷达、超声波雷达、CAN 通讯是否正常。

1.打开终端（CTRL+ALT+T），执行如下指令（在 docker 外执行）：

$ candump can0



```
nvidia@jetson-390C2D15EF46A7: ~
can0        64D   [8]  06 7D 4F A0 70 80 05 05
can0        64D   [8]  07 7A 4F A0 70 80 05 05
can0        64D   [8]  08 7D 0F A0 70 80 05 05
can0        64D   [8]  09 7D 0F A0 70 80 05 05
can0   18C4D3EF   [8]  01 00 00 00 00 00 A0 A1
can0   18C4D1EF   [8]  01 03 00 00 00 00 A0 A2
can0   18C4D6EF   [8]  01 BA 86 01 6A 18 01 00
can0   18C4D5EF   [8]  00 03 00 00 00 00 A0 A3
can0   18C4D2EF   [8]  01 FD 07 00 00 00 A0 5B
can0   18C4D9EF   [8]  01 53 01 00 00 02 50 01
can0   18C4D8EF   [8]  01 7F 80 00 00 00 E0 1E
can0   18C4D7EF   [8]  01 C8 00 C8 00 00 70 71
can0   18C4DAEF   [8]  00 00 00 00 00 00 E0 E0
can0   18C4DBEF   [8]  00 00 00 00 00 00 E0 E0
can0        303   [5]  40 FF FF FF FF
can0        302   [5]  40 FF 08 FF FF
can0        301   [5]  40 FF FF FF FF
can0   18C4D2D0   [8]  01 00 00 00 00 00 50 51
can0   18C4D3D0   [8]  01 00 00 00 00 00 70 71
can0   18C4D4D0   [8]  01 00 00 00 00 00 60 61
can0   18C4D5D0   [8]  01 00 00 00 00 00 80 81
can0   18C4D7D0   [8]  00 00 00 00 00 00 90 90
```

图 4.6　CAN 信号接收信息

有如下输出（如图示 4.6），说明系统接收到 CAN 信息，CAN 正常。18C4d5EF 一系列数据为底盘 CAN 通讯数据，18C4d5D0 为 ADAS 域控制器 CAN 通讯数据，64（ABCD）为毫米波雷达 CAN 通讯数据，301、302、303、304 为超声波雷达 CAN 通讯数据。

（五）GPS 配置向导设置及验证

1.首先进入 docker 环境，打开 dreamview 界面，根据向导完成车辆底盘、GPS 接收机、相机标定等初始化操作，点击界面顶部的 SetupWizard 按钮，界面底部会出现配置向导界面首页（如图示 4.7）

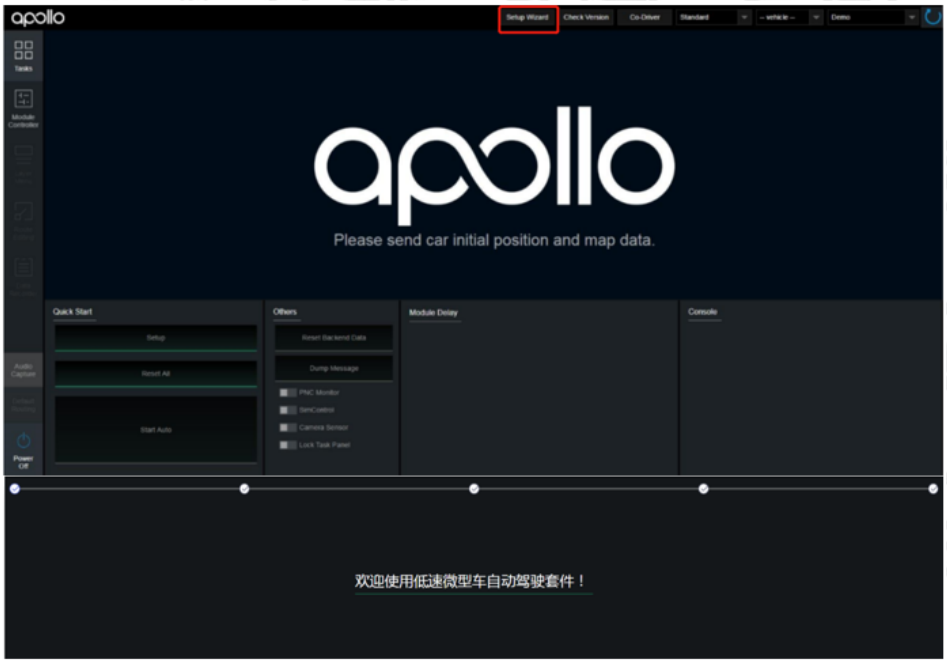


图 4.7　CAN 信号接收信息

2.点击界面任意位置进入车辆参数配置界面（如图示 4.8）

图 4.8　车辆参数配置界面

3.此处根据车辆实际参数配置，如果对参数有疑问或不明确的地方，可以将鼠标放在右侧的问号图标上，会出现字段的相应解释；配置完成后点击下一步按钮进入 GPS 接收机配置页面（如图示 4.9）

图 4.9　车辆参数配置界面

4.RTK 基站相关的配置请根据定位系统的数据信息填写；HeadOffset 需要根据定位天线和定向天线的位置确定；接收机 IP 地址需要根据 4G 路由器的 LAN 口网段配置一个未被占用的 IP 地址；杆臂值根据后天线（Primary）相对接收机的中心点位置测量；填写完成后点击

下一步按钮进入底盘校验页面（如图示 4.10）

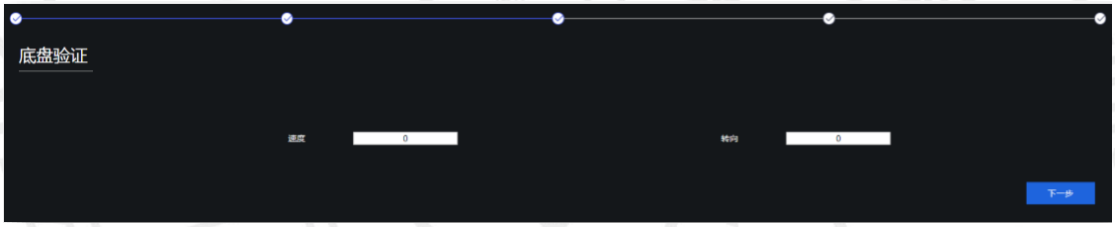


图 4.10　车辆底盘验证

5.此界面实现的功能和 teleop 程序类似，用于检测自动驾驶系统的 CANBus 模块和底盘通信是否正常；在此页面可以通过上下左右键控制底盘前进或转向，油门和转向是以百分比的形式显示在文本框中；当键盘松开按键后油门和转向会迅速下降；此页面可以直接跳过（底盘适配完成后一般会通过 teleop 验证转向和油门是否正常），点击下一步按钮进入相机校准页面（如图示 4.11）

图 4.11　相机校准

6.请按照页面所示步骤在 Windows 上运行相机标定软件完成相机校准和学习；标定成功后点击下一步按钮，进入配置完成页面（如图示 4.12）

图 4.12　配置完成

7.此时按照提示关机，再进行整车断电操作。

（六）CANBUS 通信测试

1.测试原理

（1）控制模块负责接收规划决策模块输出的行驶轨迹，通过横纵向控制算法将轨迹转换

成车辆当前位置的油门、转向和刹车信息；这些信息通过 ControlCommand（主题名称：/apollo/control）发布出来；

（2）CANBus 模块接收上游控制命令,通过 CAN 总线与车辆交互；CANBus 一方面执行控制指令，另外一方面将车辆底盘的信号反馈给上层模块；CANBus 输出的主题包括：/apollo/canbus/chassis、/apollo/canbus/chassis_detail；

（3）Teleop 工具通过模拟发送 Control 模块的 Control Command 消息来驱动 CANBus 执行转向、加减速等指令（如图示 4.13）。

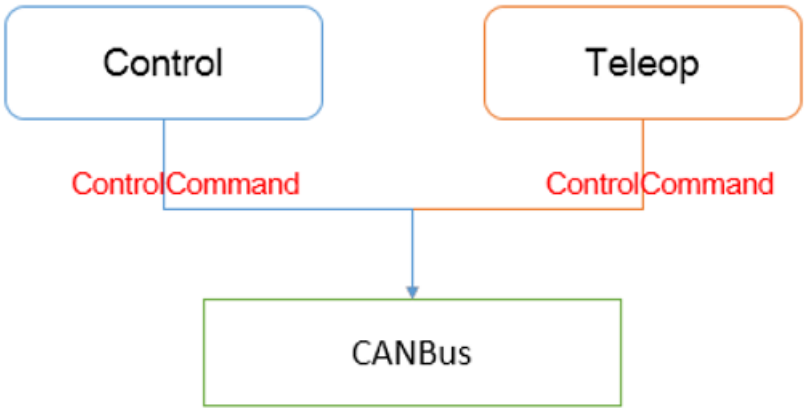


图 4.13　CANBUS 测试原理

2.测试方法

1.进入 Dreamview 界面，点击 CanBus、Guardian 按钮，启动 CANBUS（如图示 4.14）。

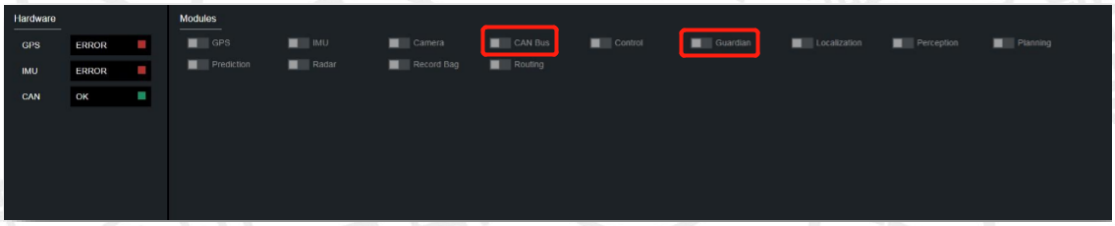


图 4.14　启动 CANBUS 和 Guardian

2.终端下执行如下指令检测 CANBUS 连接：

$ rostopic echo /apollo/canbus/chassis

3.指令执行后，左右推动遥控器摇杆，使车轮发生发生转动，若看到终端有数字发生变化，则 CANBUS 连接正常。（如图示 4.15）

```
Chassis
  engine_started: True
  speed_mps: 0.00000
  throttle_percentage: 0.00000
  brake_percentage: 0.00000
  steering_percentage: 0.00000
  driving_mode: COMPLETE_MANUAL
  error_code: NO_ERROR
  gear_location: GEAR_REVERSE
  header:
    timestamp_sec: 1551682297.83584
    module_name: chassis
    sequence_num: 10234
  engage_advice:
    advice: READY_TO_ENGAGE
```

图 4.15　遥控器操作终端数据

4.首先确保底盘 CAN 总线和计算单元之间连接正确，然后终端切换到 apollo 目录下启动键盘控制，执行指令如下（如图示 4.16）

$ ./bazel-bin/modules/canbus/tools/teleop

或

$ bash scripts/canbus_teleop.sh

```
eleop:
eading from keyboard now.
--------------------------
se arrow keys to drive the car.
====================    KEYBOARD MAP    ====================
ELP:                [h]     |
et Action      :    [m]+Num
                     0 RESET ACTION
                     1 START ACTION

-------------------------------------------------------------
et Gear:            [G]+Num
                     0 GEAR_NEUTRAL
                     1 GEAR_DRIVE
                     2 GEAR_REVERSE
                     3 GEAR_PARKING
                     4 GEAR_LOW
                     5 GEAR_INVALID
                     6 GEAR_NONE

-------------------------------------------------------------
hrottle/Speed up:   [W]     |  Set Throttle:       [T]+Num
rake/Speed down:    [S]     |  Set Brake:          [B]+Num
teer LEFT:          [A]     |  Steer RIGHT:        [D]
arking Brake:      [P]     |  Emergency Stop      [E]

-------------------------------------------------------------
```

图 4.16　键盘控制终端

5.根据提示按 M0 重置系统，按 M1 是开始控制底盘。 按 G1，挂前进档，按几次 A 或者

D，看看车轮是否转动；按几次 W，看车辆是否前进，按几次 S 看车辆是否停下来。请小心测试，不要轻易长时间连续按键，以防车辆突然高速动作发生事故。

（七）激光雷达配置、检测

设置激光雷达目的 IP 与自动驾驶处理器 IP 一致，保证自动驾驶处理器和激光雷达通讯数据正常。以本地的笔记本或者台式机及其他设备在接收数据时，需将 IP 设置为雷达的目的 IP（默认为：192.168.1.102）。用户在设置雷达 IP 时，本地 IP 与目的 IP 请勿设置为同一 IP，否则雷达将不能正常工作。

连接激光雷达时，电脑与激光雷达的 IP 在不同网段时，需要设置网关；相同网段时，设置不同 IP 即可，例如：192.168.1.102，子网掩码为 255.255.255.0。

1.将激光雷达与电脑使用网线相连，并给激光雷达供电。

2.选择激光雷达的以太网，使用电脑连接激光雷达的以太网（如图示 4.17）。

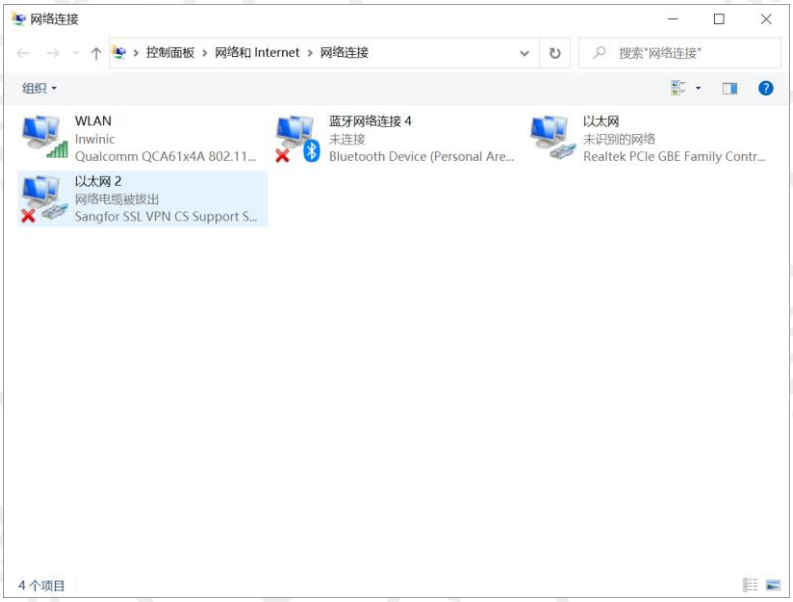


图 4.17　以太网配置

3.查看以太网的详细信息，网络连接详细信息 IP 地址，如 192.168.1.102（如图示 4.18）。

图 4.18　以太网详细信息

4.电脑安装 LSC16-[Client]200902_v3.0.4 软件，并打开 LSC16-[Client]200902_v3.0.4 软件（如图示 4.19、4.20）。

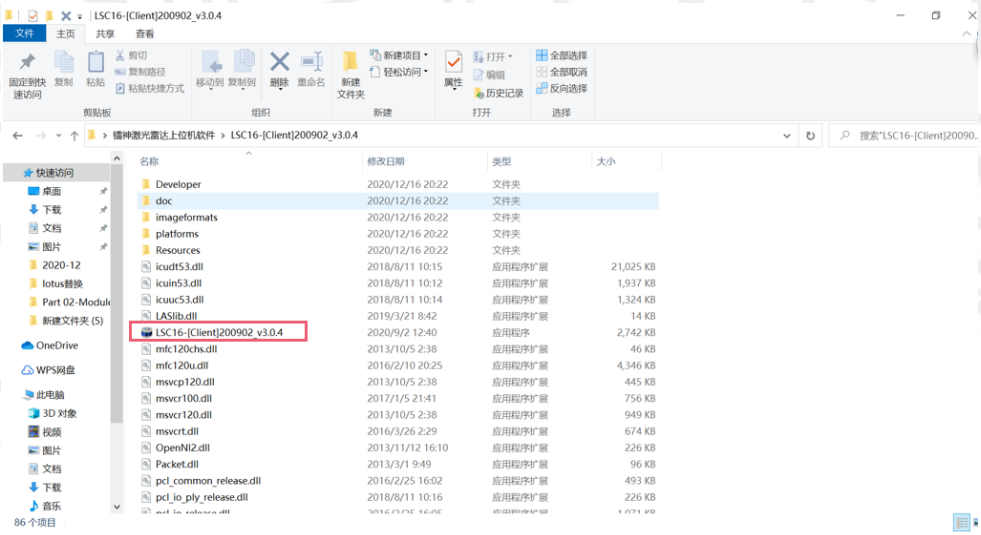


图 4.19　安装 LSC16-[Client]200902_v3.0.4 软件

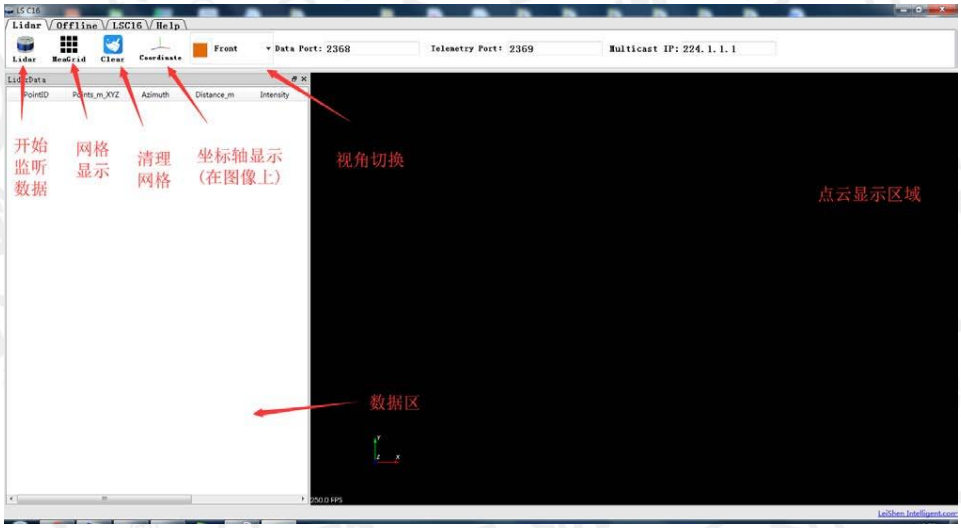


图 4.20　软件功能介绍

5.打开软件应用程序，点击 Lidar 选项选择 LSC16-1（单次回波）选项（如图示 4.21）。

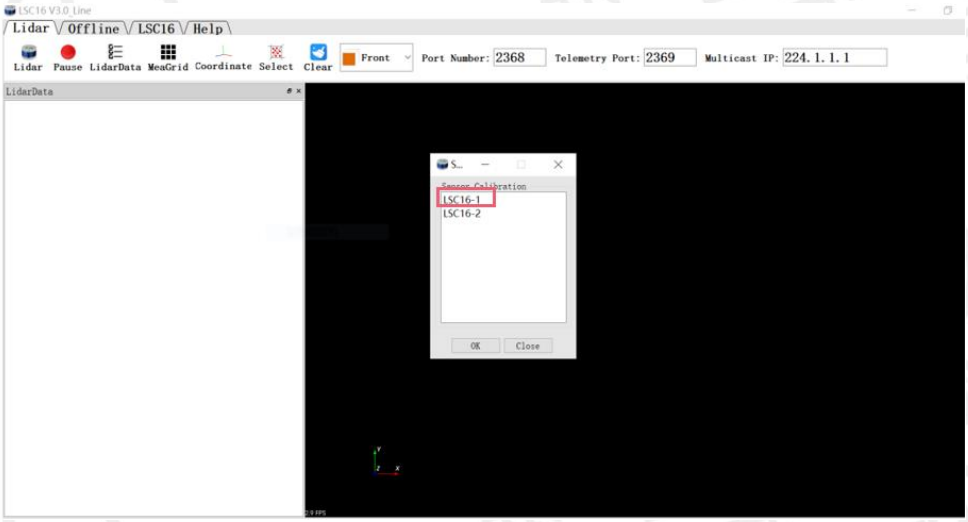


图 4.21　选择单次回波选项

6.选择 LSC16-1 选项之后，会看到激光雷达数据信息（如图示 4.22）。

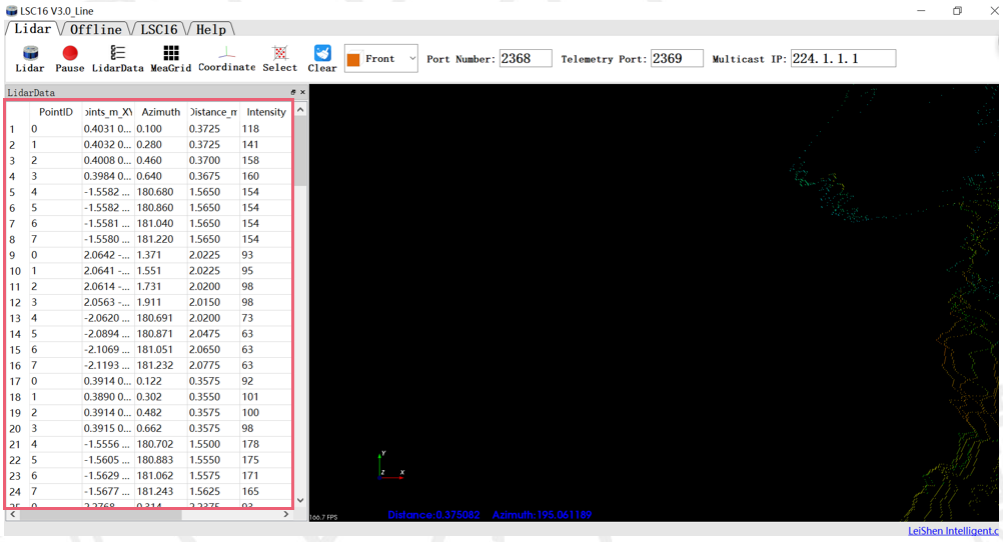


图 4.22　激光雷达数据信息

7.选择 LSC16 选项，点击 Parameter 对雷达目的 IP 进行修改(如图示 4.23）。

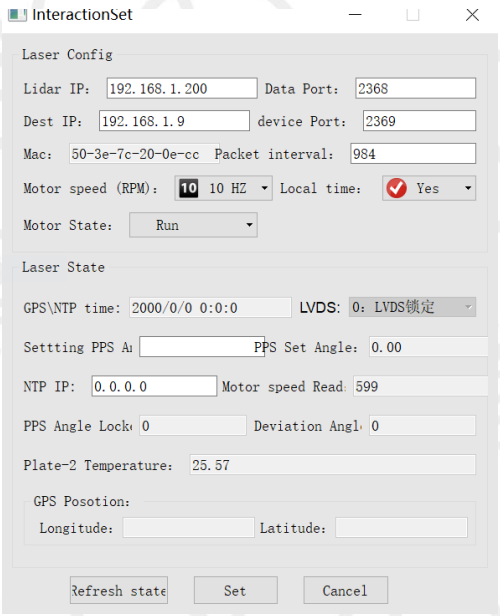


图 4.23　Refresh state 设置完成

点击状态信息（Refresh state）刷新，可获取雷达的之前的配置信息（设备信息流内容），窗体上部为雷达参数设置部分，可以设置的参数包括雷达本地 IP，雷达目的 IP（Dest IP 更改为 192.168.1.9),雷达转速设置(在 combobox 下可选择 5hz/10hz/20hz 三种转速模式,将雷达频率更改为 10hz)。点击设置按钮，需断电之后，使设置生效。

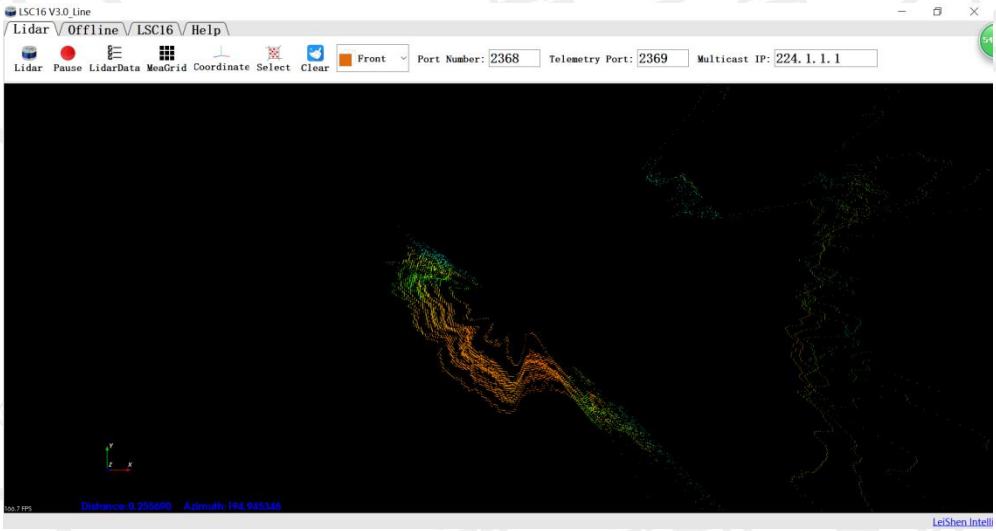


图 4.24　激光雷达功能正常

将激光雷达参数设置完成后，将激光雷达安装在智能网联汽车上，正确连接路由器和电源

后，可以使用 ping 指令测试激光雷达是否与计算单元通讯，正确结果如下（如图示 4.25），如果不能 ping 通，请检查激光雷达及 IPC 的 ip 地址设置。

8.激光雷达的默认 IP 地址是 192.168.1.200，ping 指令如下：

$ ping 192.168.1.200

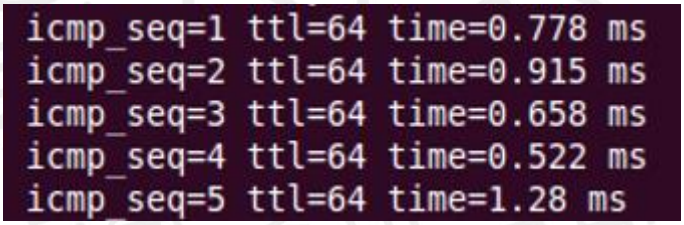


图 4.25 键盘控制终端

9.启动激光雷达：进入 docker 后，执行 lidar 驱动脚本：

$ ./scripts/lslidar_c16.sh

或

$ bash scripts/lslidar_c16.sh

10.检查激光雷达是否正常启动，是否有点云数据输出（如图示 4.26），执行如下指令：

$ rostopic echo /lslidar_point_cloud

图 4.26 激光雷达话题数据

激光雷达话题数据是激光点所在空间坐标位置，终端有数据输出，则激光雷达已正常启动。

11.查看激光雷达点云图：首先进入 Dreamview 界面，启动 Lslidar 16 模块。

12.进入 docker 环境，打开 Dreamview 界面。在 Tasks 标签栏下打开 Sim Control（如图示 4.27）。

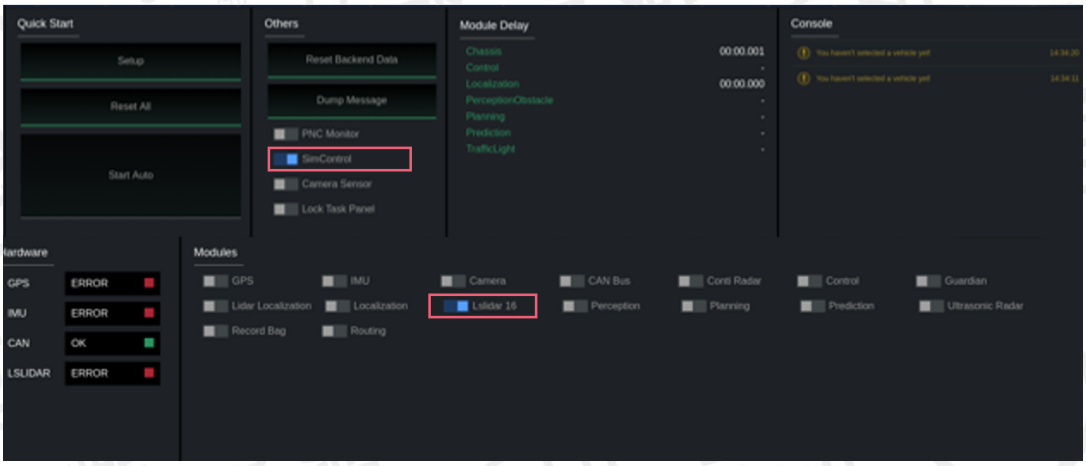


图 4.27　打开激光雷达模块

13.界面中出现车辆模型后，点击 Layer Menu 按钮。

14.进入 Layer Menu 对应的界面后，点击 PointCloud，则会输出激光雷达点云图（如图示 4.28、4.29）。

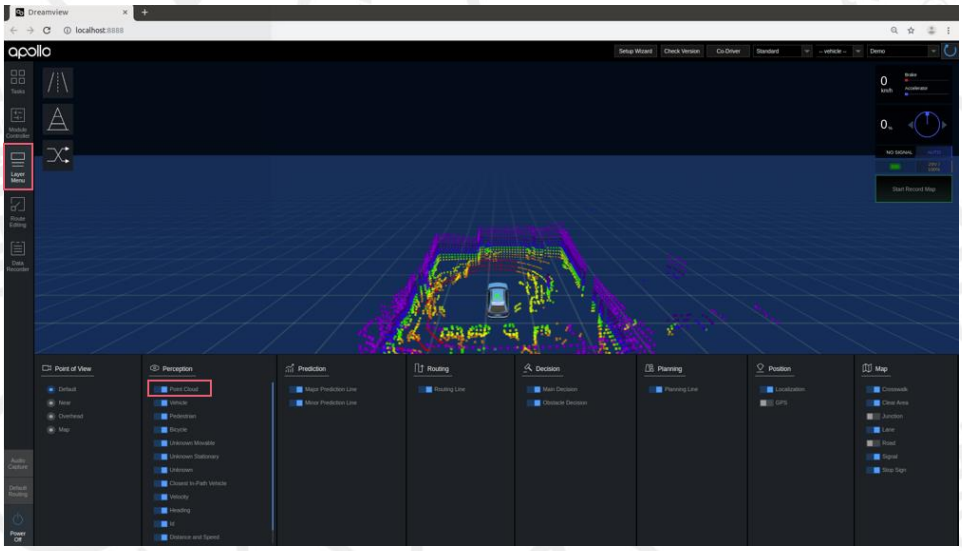


图 4.28　Layer Menu 界面

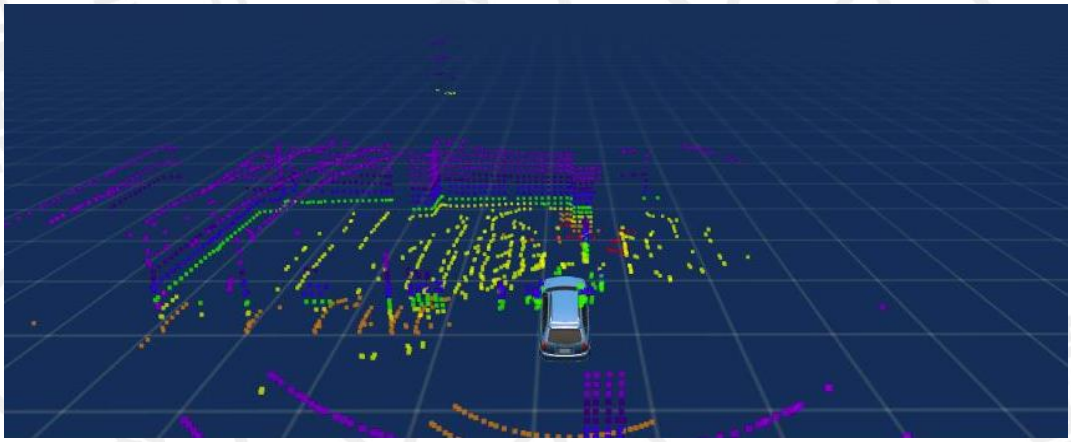


图 4.29 激光雷达点云图

激光雷达点云图是对激光雷达周围物体进行扫描的结果，通过激光雷达对结果位置的信息获取，并且呈现不同颜色来显示位置信息。

（八）毫米波雷达配置、检测

1.启动毫米波雷达：进入 docker 后，执行 conti_radar 驱动脚本：

$ ./scripts/conti_radar.sh

或

$ bash scripts/conti_radar.sh

此外，也可以进入进入 dreamview 界面，点击 conti radar 按钮启动。（如图示 4.30）

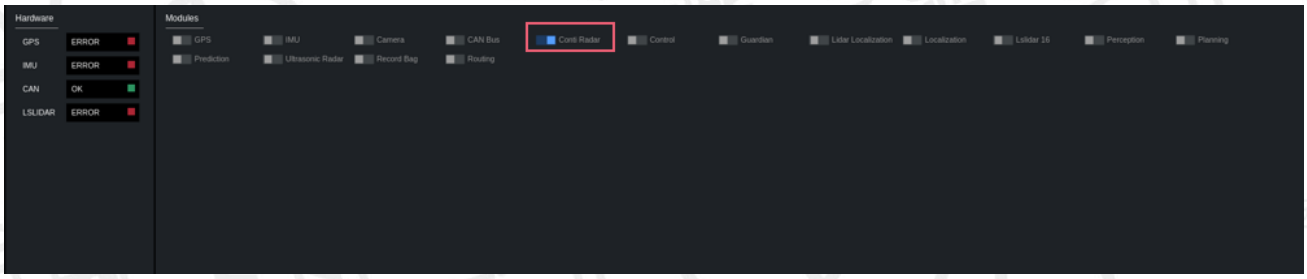


图 4.30　启动 conti radar

2.输入以下命令，打开该话题，将会输出目标物体的纵向、横向数据：

$ rostopic echo /apollo/sensor/conti_radar

智能网联整车综合应用软件，输出内容（如图示 4.31）

```
contiobs {
  header {
    timestamp_sec: 1555915766.14
    module_name: "conti_radar"
    sequence_num: 4938
  }
  clusterortrack: false
  obstacle_id: 66
  longitude_dist: 112.4
  lateral_dist: 3.2
  longitude_vel: 0.0
  lateral_vel: -4.75
  rcs: 6.0
  dynprop: 1
  longitude_dist_rms: 0.371
  lateral_dist_rms: 0.616
  longitude_vel_rms: 0.478
  lateral_vel_rms: 0.794
  probexist: 0.999
  meas_state: 2
  longitude_accel: -0.04
  lateral_accel: 0.0
  oritation_angle: 0.0
  longitude_accel_rms: 1.023
  lateral_accel_rms: 0.005
  oritation_angle_rms: 1.909
  length: 2.0
  width: 2.6
  obstacle_class: 1
}
object_list_status {
  nof_objects: 55
  meas_counter: 10000
  interface_version: 0
}
```

图 4.31　毫米波雷达纵向横向数据

毫米波雷达话题数据参数主要为 longitude_dist 横向距离，lateral_dist 纵向距离，longitude_vel 横向速度，lateral_vel 纵向速度。

（九）IMU 配置、检测

1.启动 IMU：进入 docker 后，执行 IMU 驱动脚本（效果如图示 4.32）

$ ./scripts/imu_100d2.sh

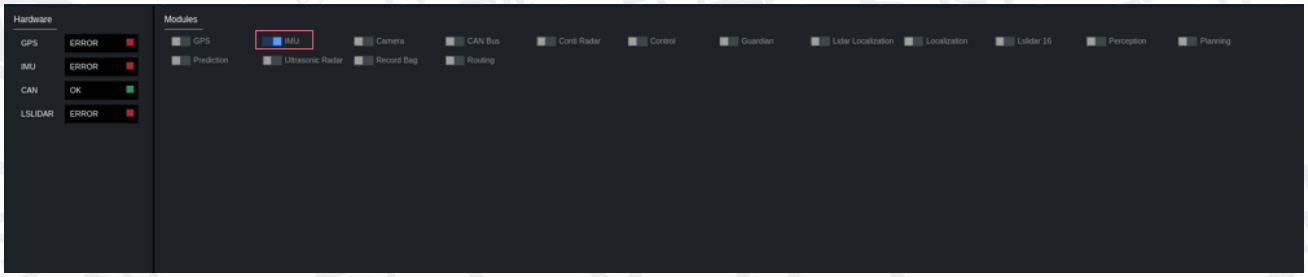


图 4.32 启动 IMU

2.命令行显示超声波雷达信号数据及数据显示界面（效果如图示 4.33）

$ rostopic echo /imu_data

```
header:
  seq: 190
  stamp:
    secs: 1600830757
    nsecs: 700643218
  frame_id: IMU_link
orientation:
  x: 0.0087265222071
  y: -0.00174526193515
  z: 1.52306701683e-05
  w: 0.999960400035
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: -0.000532112589514
  y: 0.0
  z: -0.000532112589514
angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
  x: 0.0227526855469
  y: 0.168848876953
  z: 9.67168762207
```

图 4.33 IMU 话题数据

imu 话题数据参数主要是 orientation 位置参数(四元数），angular_velocity 角速度，linear_acceleration 线性加速度。

（十）GPS 接收机配置、检测

1.启动 GPS 接收机：进入 docker 后，执行 gps 驱动脚本（效果如图示 4.34）

$ ./scripts/gps.sh

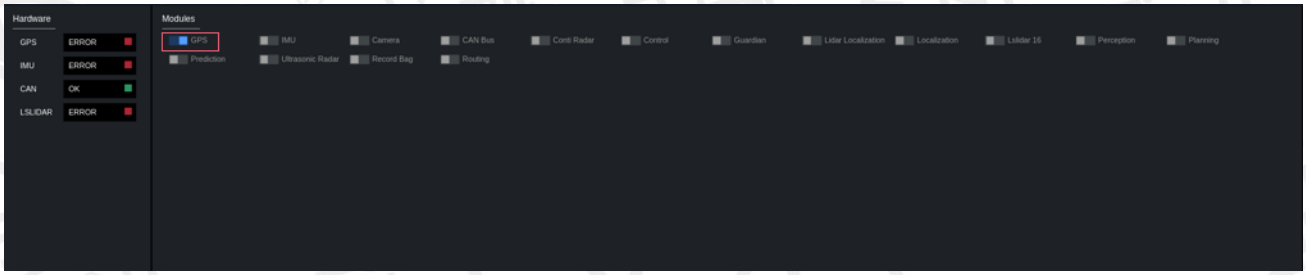


图 4.34 启动 gps

输入以下命令，打开该话题，将会输出显示 GPS 信号数据（效果如图示 4.35）

$ rostopic echo /apollo/sensor/gnss/best_pose

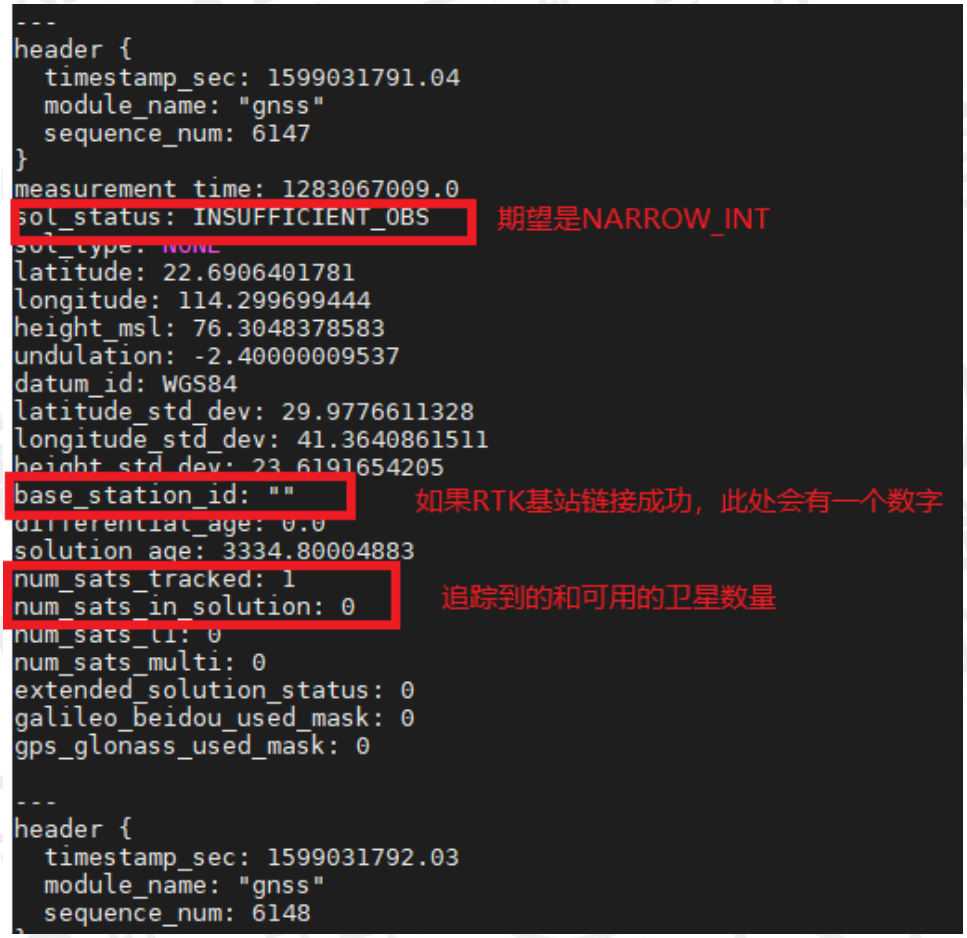


图 4.35 GPS 接收机话题数据

GPS 接收机话题数据输出为固定解状态（NARROW_INT），证明接收机正常工作。

（十一）超声波雷达配置、检测

1.启动超声波雷达：进入 docker 后，执行 ultrasonic 驱动脚本（效果如图示 4.36）

$ ./scripts/ultrasonic_radar.sh

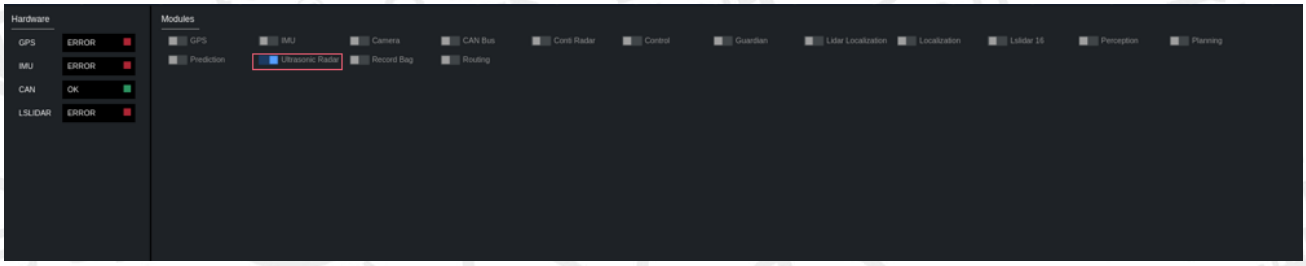


图 4.36 启动 Ultrasonic Radar

2.输入以下命令，打开该话题，将会输出显示超声波雷达信号数据（效果如图示 4.37）

$ rostopic echo /apollo/sensor/ultrasonic_radar

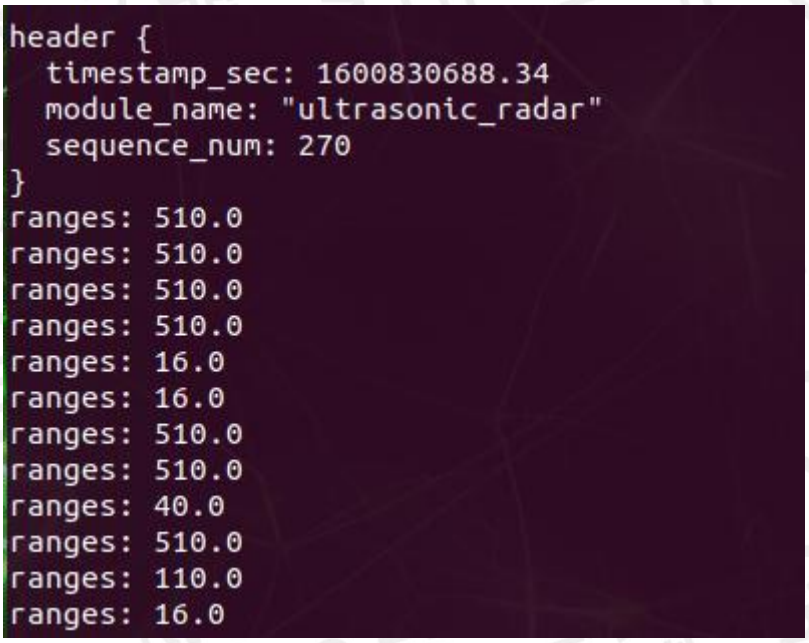


图 4.37 超声波雷达话题数据

超声波雷达话题数据参数显示的是障碍物距离超声波雷达距离，本车辆超声波雷达探测有效距离为 510cm，如果在 510cm 内没有障碍物全部显示 510，如果有显示障碍物距超声波雷达距离。

（十二）定位模块配置、检测

INS 利用安装在载体上的惯性器件敏感载体的运动，输出载体的姿态和位置信息，具有很强的自主性、保密性、灵活性且机动性强，具备多功能参数输出，但是由于导航精精度随时变化，INS 不能长时间单独工作，必须连续校准。GNSS 由于需要接受足够数量的卫星信号才能够实现定位，受各种物理、电磁信号等遮挡影响比较大。

智能网联汽车通过 GPS+惯导进行定位，并实时的准确的传输车辆位置信号。

1.启动定位模块：进入 docker 后，执行 localization 驱动脚本（效果如图示 4.38）

$ ./scripts/localization.sh

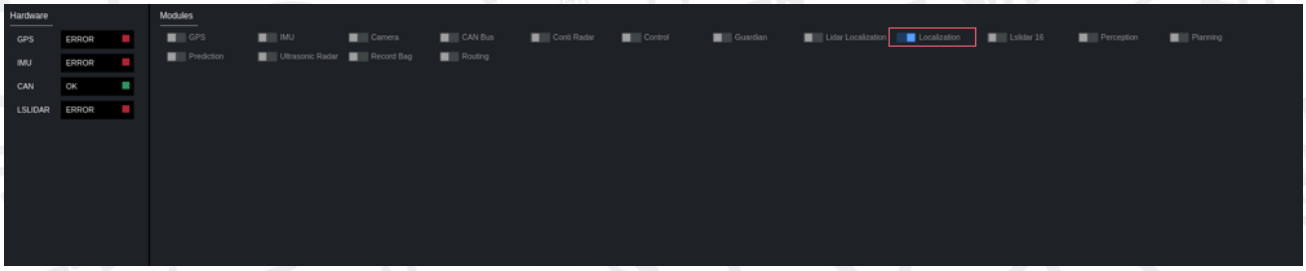
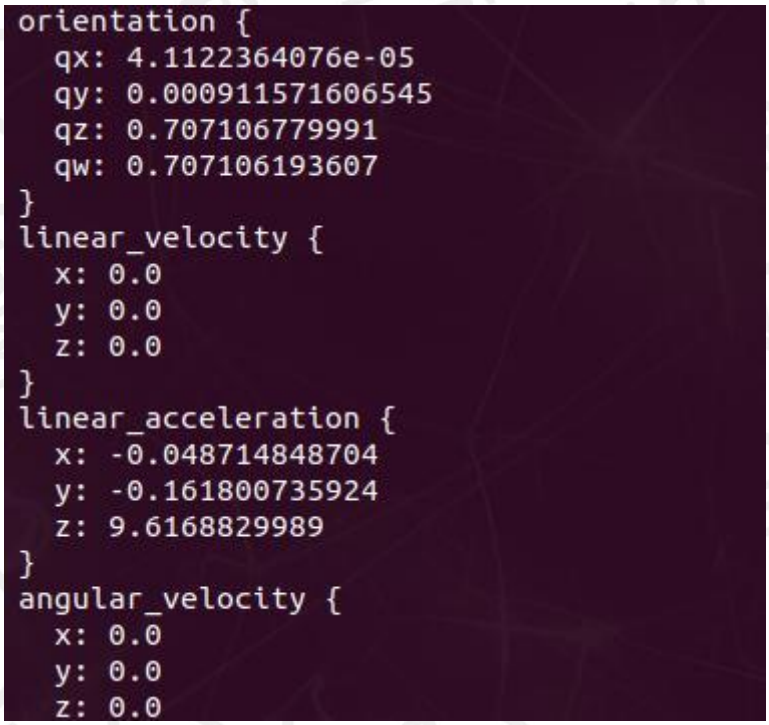


图 4.38　启动 localization

2.输入以下命令，打开该话题，将会输出显示定位模块信号数据（效果如图示 4.39）

$ rostopic echo /apollo/localization/pose



```
orientation {
  qx: 4.1122364076e-05
  qy: 0.000911571606545
  qz: 0.707106779991
  qw: 0.707106193607
}
linear_velocity {
  x: 0.0
  y: 0.0
  z: 0.0
}
linear_acceleration {
  x: -0.048714848704
  y: -0.161800735924
  z: 9.6168829989
}
angular_velocity {
  x: 0.0
  y: 0.0
  z: 0.0
```

图 4.39　定位模块话题数据

定位模块话题数据参数 orientation 定位信息，linear_velocity 线性速度，linear_acceleration 线性加速度，angular_velocity 角速度等。

（十三）双目摄像头检测及标定

1.双目摄像头检测

1.1 启动摄像头：进入 docker 环境，打开 dreamview 界面。在 Tasks 标签栏下依次打开 Sim Control 和 Camera Sensor 开关，Camera Sensor 打开后，关闭 SimControl（效果如图示 4.40）。

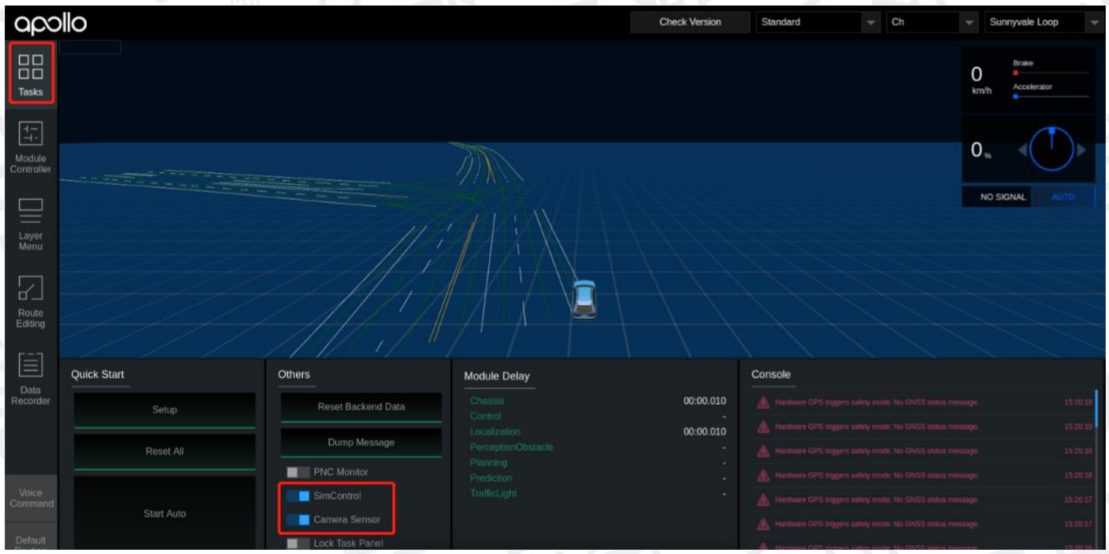


图 4.40 打开 Sim Control 和 Camera Sensor 开关

1.2 在 Module Controller 标签栏下，打开 Camera 开关，（效果如图示 4.41）

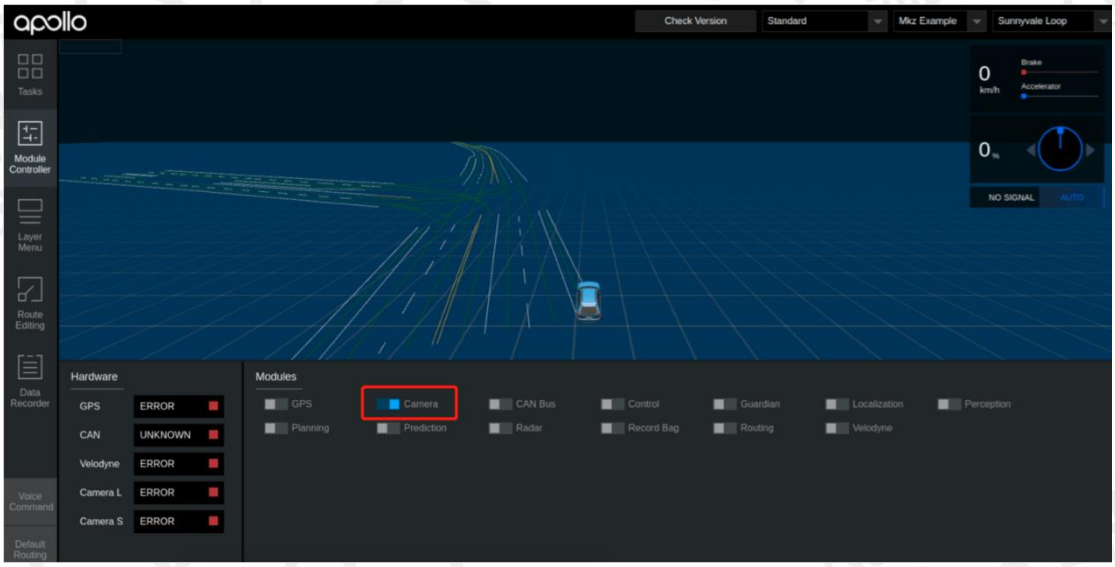


图 4.41 打开 Camera 开关

1.3 在 dreamview 左上角将会出现摄像头采集的图像（效果如图示 4.42）

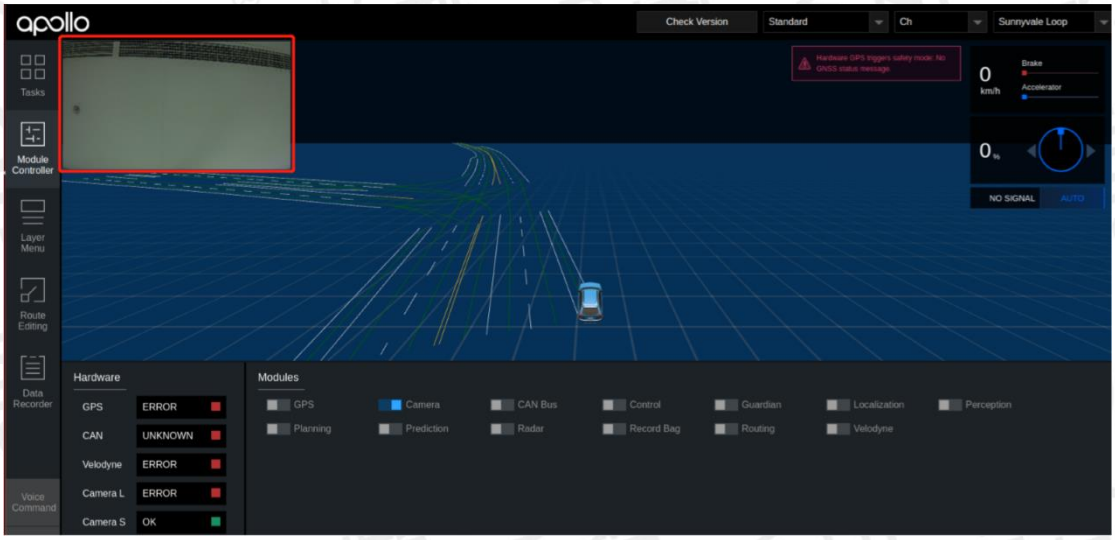


图 4.42 摄像头采集的图像

2.双目摄像头标定

2.1 使用提供的账号和密码登录 SmarterEye 安装工具（FieldHelperSetup.exe）（效果如图示 4.43）。

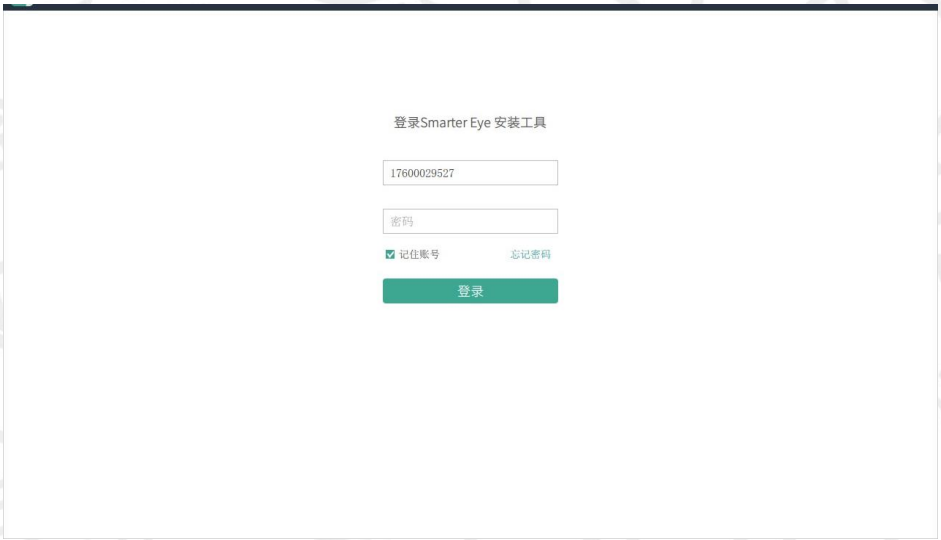


图 4.43 登录 SmarterEye 安装工具

2.2 打开 SmarterEye 安装工具（FieldHelperSetup.exe），选择连接设备（效果如图示 4.44）。

图 4.44　打开 SmarterEye 安装工具连接设备

2.3 配置 PC 设备的 IP 与产品设备的 IP 在同一网段内，192.168.1.251 除外，其他 192.168.1.XX 均可使用，输入设备 IP（192.168.1.251），点击【确定】按钮，进入 SmarterEye 安装工具（FieldHelperSetup.exe）主页（效果如图示 4.45、4.46）。

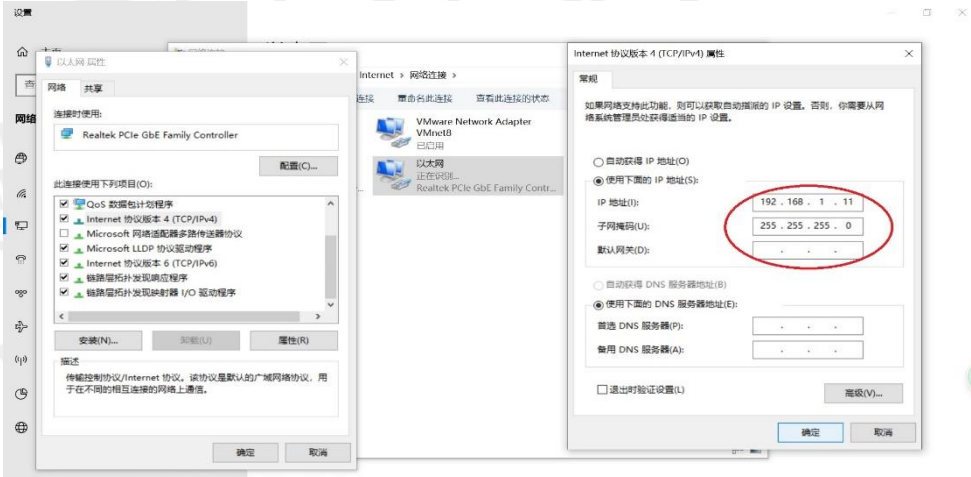


图 4.45　SmarterEye 安装工具主页

图 4.46　输入 IP 地址

2.4 进入到主页后，显示设备的型号、序列号、固件类型、设备状态、固件版本以及显示

屏的型号、序列号、固件版本，点击【安装设备】按钮，进入设备安装流程（效果如图示 4.47）。

图 4.47　显示设备参数

2.5 在相机检测中，调节车辆和摄像头位置使镜头画面在车道中央，然后选择左下角跳过该步骤（效果如图示 4.48）。

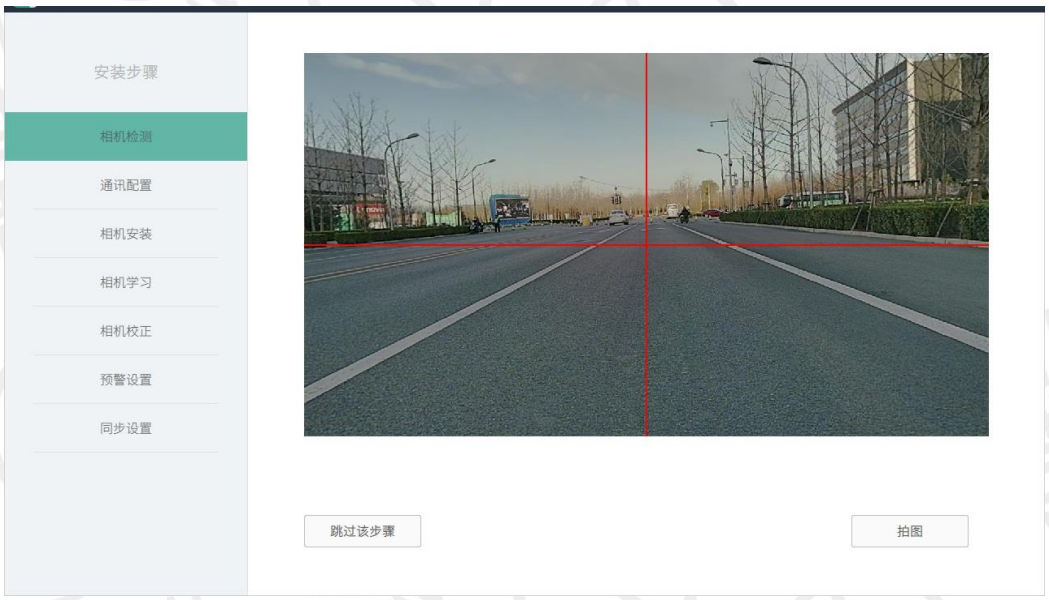


图 4.48　相机检测

2.6 在通讯设置中，根据现场实际需求去配置车速参数，点击保存并下一步。

此页面所有可选项目设置为不需要第三方通讯接口，不使用车速信号，此类数据隔绝选项（效果如图示 4.49）。

图 4.49　通信配置

2.7 在相机安装中，将相机靶标横向放置于三脚架，靠近车头靶标垂直于地面，且靶标竖向中线（示意图蓝线）放置于车辆中轴线位置，靶标横向中心线（示意图红线）与相机等高（示意图绿线）；通过调节设备后面的螺旋钮，使用 PH1 的十字螺丝刀松动中间的螺丝后来调整相机的俯仰角，相机视野内，中间的黄线与靶标中心线重合，然后根据实际安装位置填入相机的安装参数，点击保存并下一步按钮（效果如图示 4.50、4.51、4.52、4.53）。

图 4.50 相机安装及调整

图 4.51 相机旋转钮

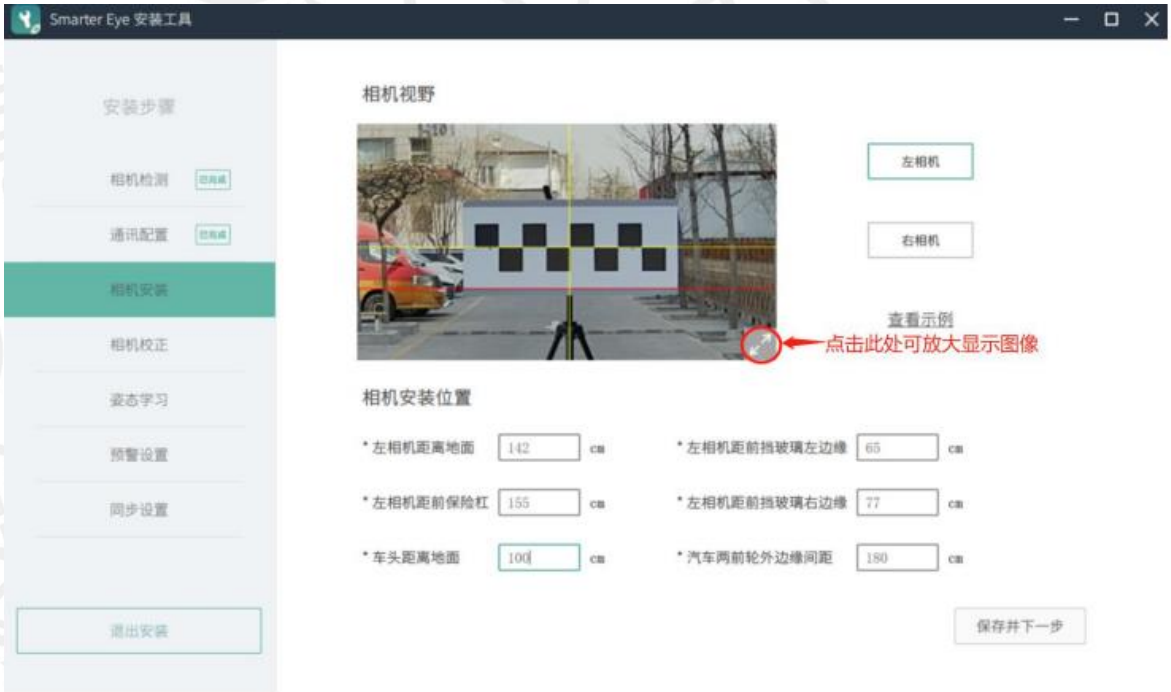


图 4.52 相机安装参数设置

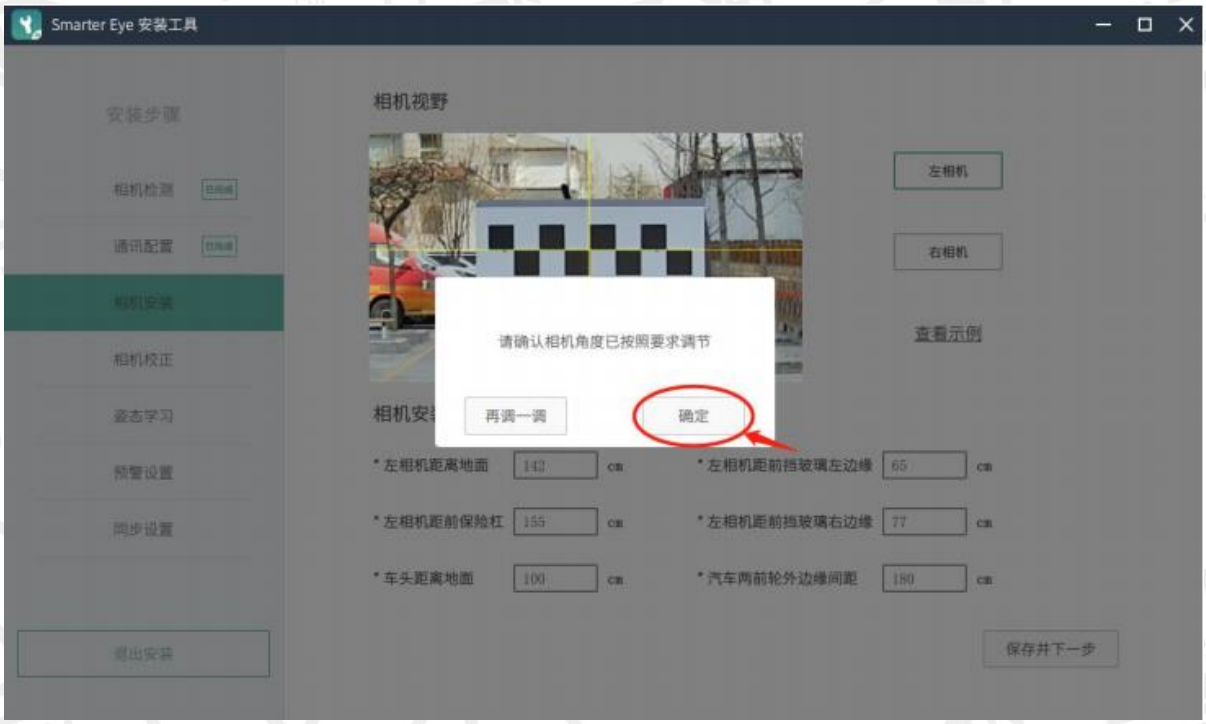


图 4.53　相机角度调整

操作者需要实际测量和输入相机安装位置信息的说明：

a) 车辆左侧相机距离地面的距离（0-300 之间的整数，单位：cm）；

b）车辆左侧相机距前挡风玻璃左边缘的距离（30-250 之间的整数，单位：cm）；

c）车辆左侧相机距前保险杠的距离（0-250 之间的整数，单位：cm）；

d）车辆左侧相机距前挡风玻璃右边缘的距离（30-250 之间的整数，单位：cm）；

e）车头距离地面的距离（0-300 之间的整数，单位：cm）；

f）车辆两个前轮外边缘的间距（100-500 之间的整数，单位：cm）

提示：

车头距离地面的距离测量位置是前挡风玻璃的下沿到地面的垂直距离。

2.8.在相机校正中，减小相机图的畸变影响，需用 90 厘米*40 厘米的黑白棋盘格靶标进行相机标定。相机按 4 米/6 米/8 米/10 米的顺序在相机前十字红框内的地方拍摄四组图像，使用卷尺测量的距离误差在±20 厘米内即可。拍图时，需将靶标与实时图像中的参考十字线的中心重合使红框自动变为绿框后进行拍图（效果如图示 4.54、4.55、4.56、4.57）。

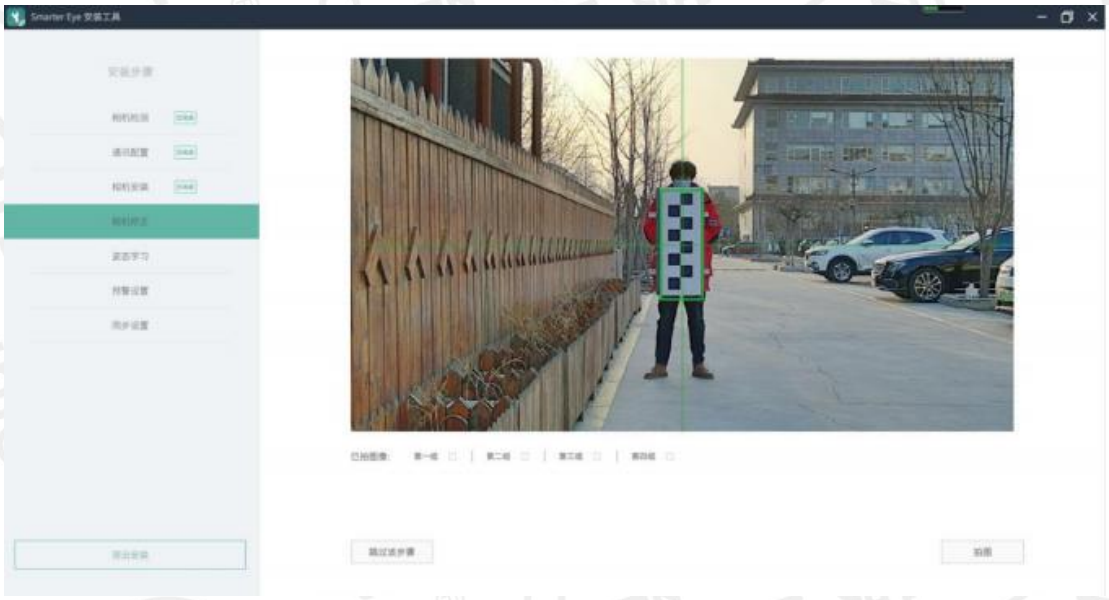


图 4.54 相机校正 1

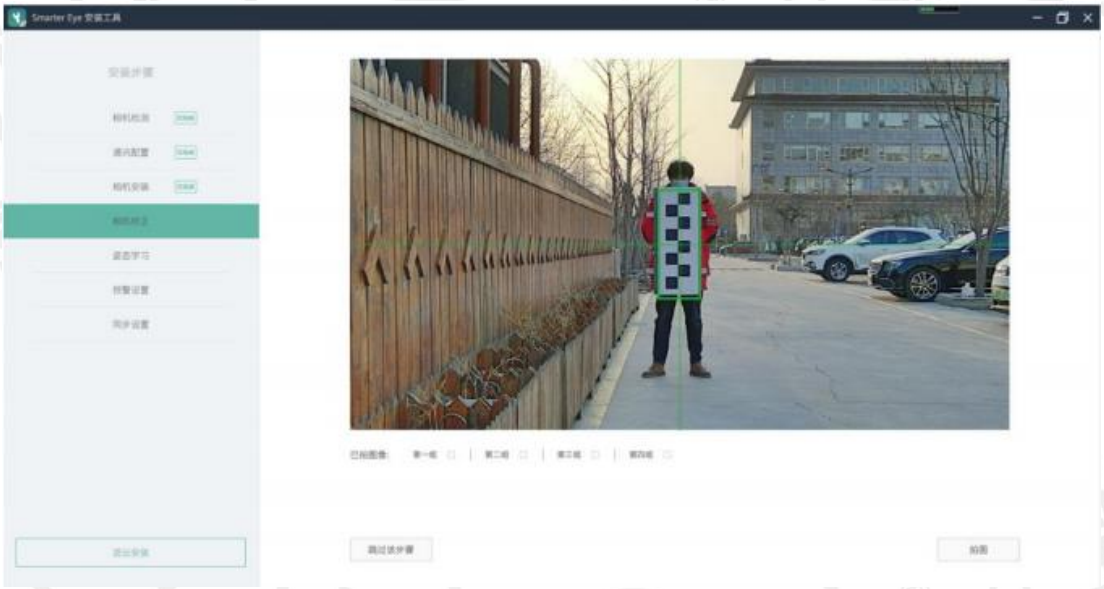


图 4.55 相机校正 2

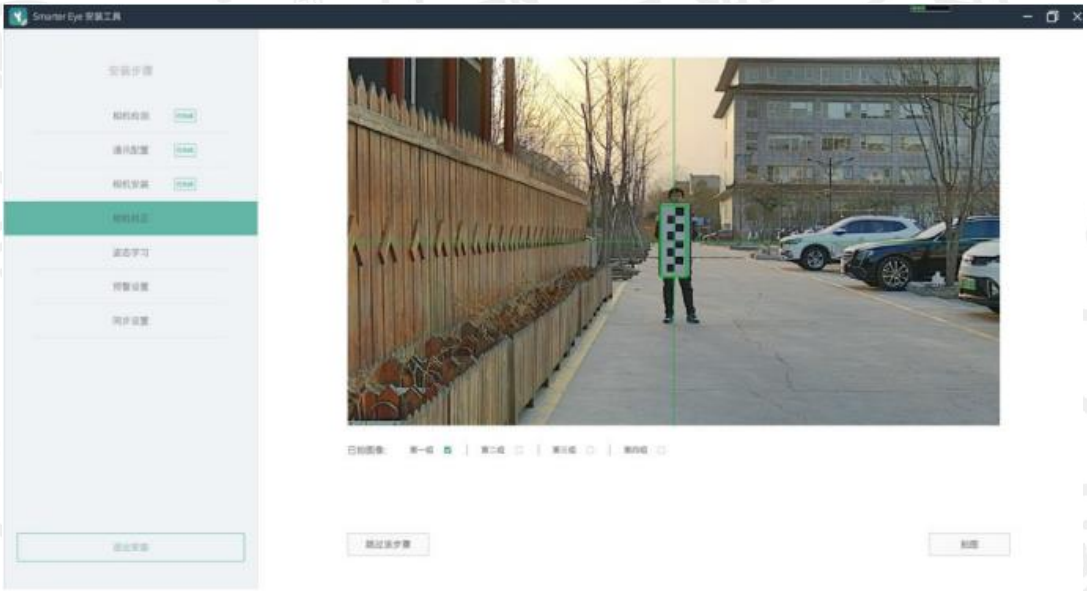


图 4.56　相机校正 3

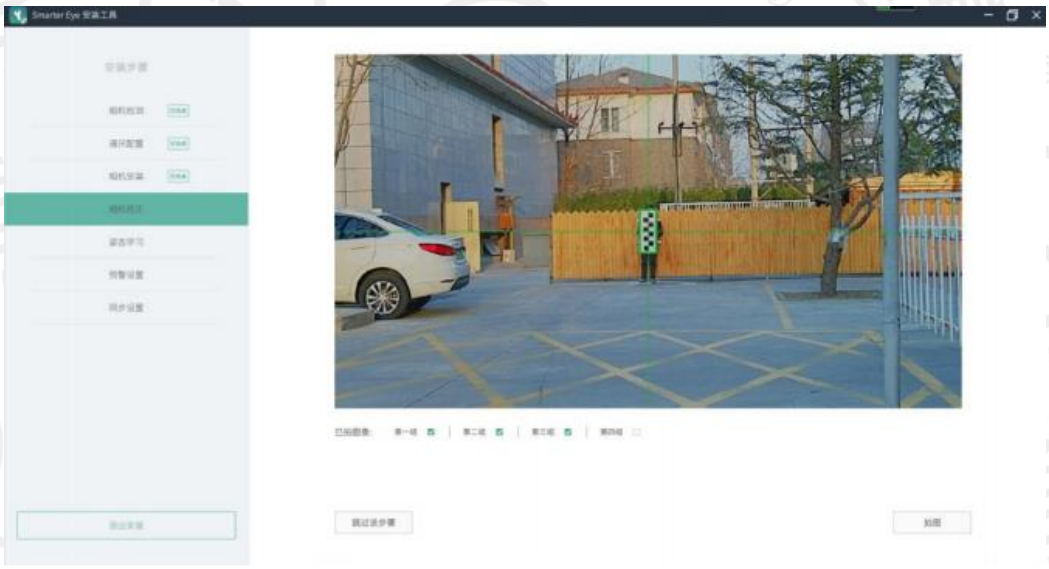


图 4.57 相机校正 4

2.9 设置姿态学习，选择靶标学习，进行靶标学习（效果如图示 4.58、4.59、4.60、4.61）。

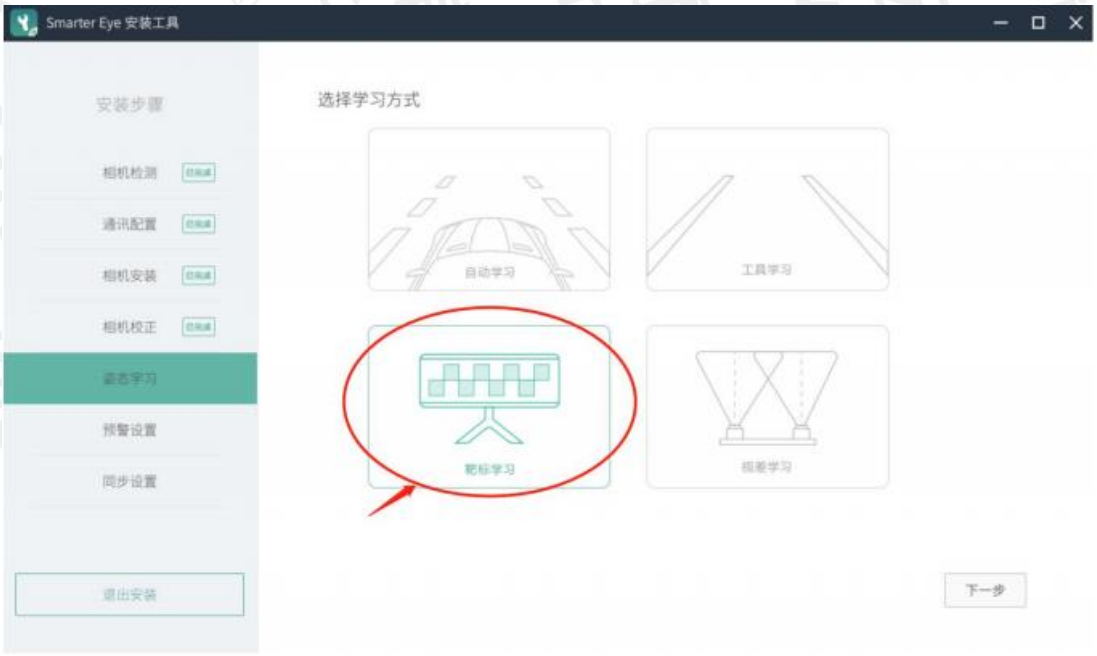


图 4.58　设置姿态学习

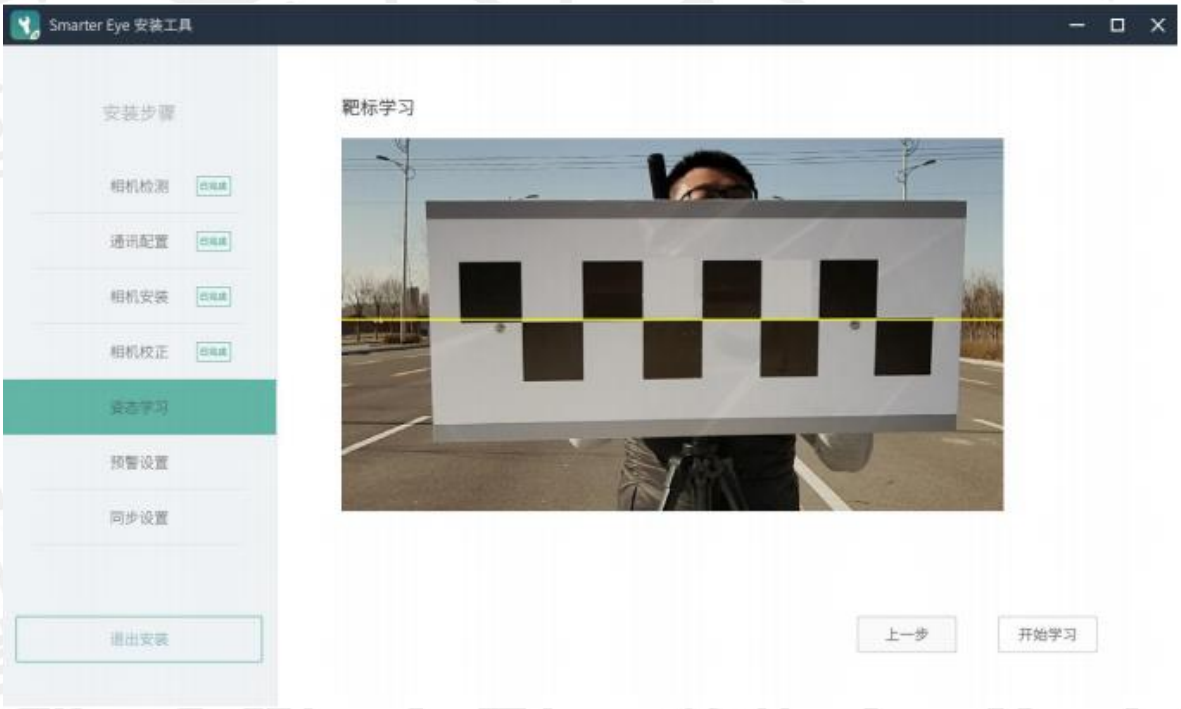


图 4.59　靶标学习

2.10 在预警设置中，用户可以根据特定需求，可将预警设置的“启用车速”参数进行适当的修改（默认值即可），点击保存并下一步。

图 4.60　设置预警设置参数

2.11 在同步设置中，使用系统预留参数，点击同步设置，设备会进行安装并重启。

图 4.61　设置配置成功

2.12 如果由于设备安装位置、俯仰角度、通讯配置等，需要重新进行设备安装，点击恢复出厂设置，设备会自动重启，重新运行安装工具即可重新进行设备安装（效果如图示 4.62）。

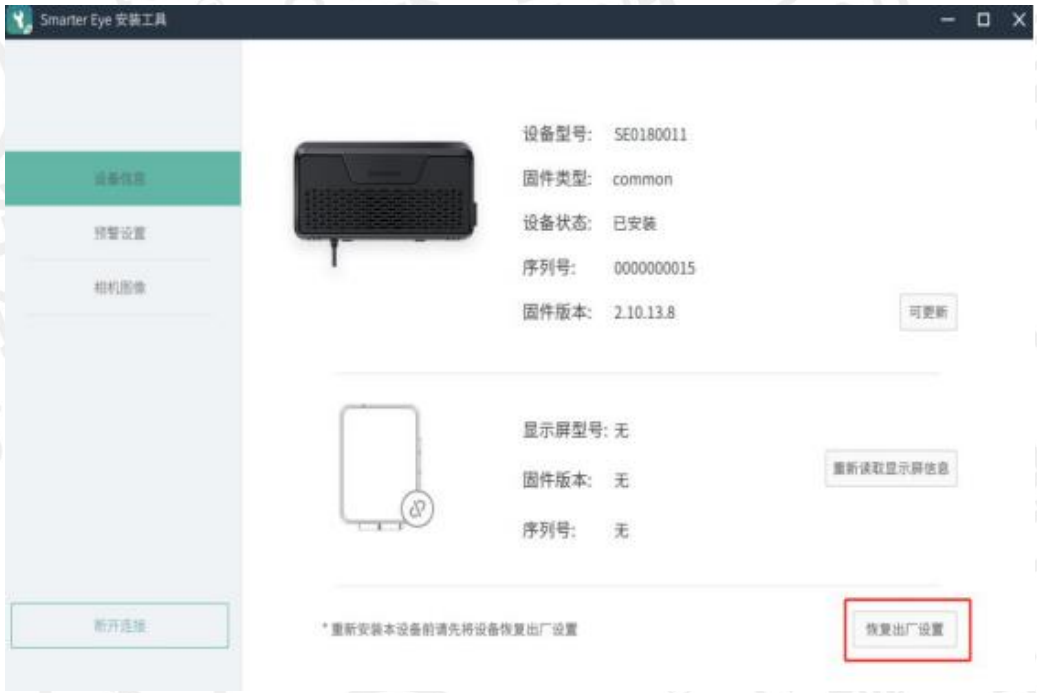


图 4.62 设备信息中恢复出厂设置

（十五）3D-AVM360 环视全景标定

ADAS 通过雷达、摄像头等环境感知传感器获取行车环境信息，并将这些信息通过图像和语音等方式传递给驾驶人，从而提高驾驶人视野较差环境下驾驶车辆的安全性。3D-AVM360 环视全景标定步骤：

1.连接好 ADAS 域控制器及平板电脑。

2.在平板电脑上进行 360 全景标定（效果如图示 4.63）。

3.点击红框设置进入标定工程界面。

图 4.63 点击设置进入标定工程界面

4.点击“关于”界面，显示如上图所示，点击“开始标定”按钮，进行标定（效果如图示 4.64）。

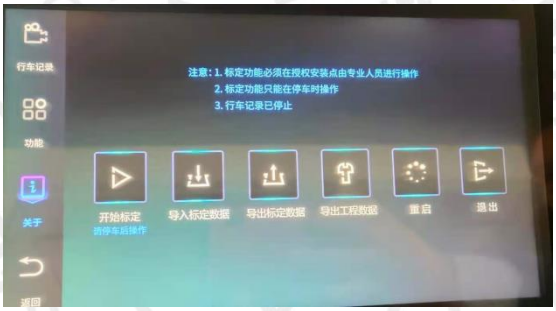


图 4.64　选择关于界面开始标定

5.标定密码输入 1234，进入标定工程界面（效果如图示 4.65）。

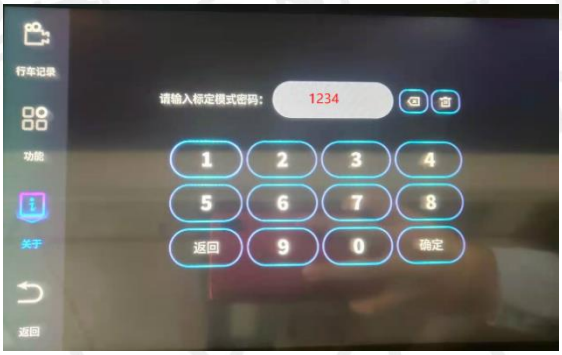


图 4.65　输入标定密码

6.利用测量工具，按照图示标定流程，正确测量实际合理的标定车长、标定车宽、轴距、后悬、标定布尺寸规格等参数，数据填写记录完成后，点击镜头参数按钮（效果如图示 4.66、4.67）。

图 4.66　输入标定密码

图 4.67 镜头参数配置

镜头参数选项提供两种不同型号的摄像头参数，具体差异表现在摄像头的镜头畸变系数、摄像头内参及摄像头 Sensor 型号区别，选择合适摄像头并记录工单，以免影响标定效果，摄像头型号选型好后进入手动标定流程。

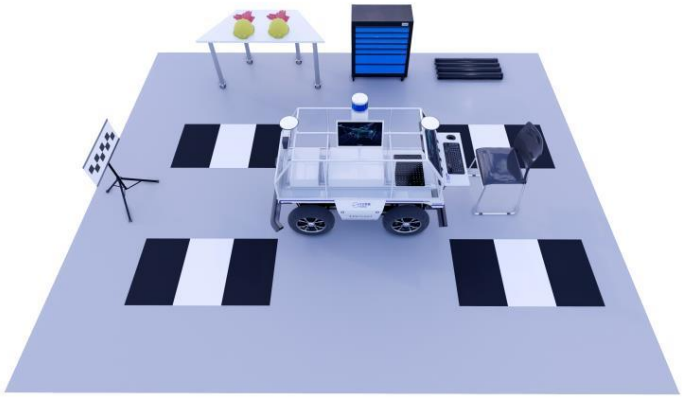


图 4.68 场地布置标准

图 4.69 手动标定流程

（效果如图示 4.68、4.69），按照操作规范要求，根据车辆尺寸，在距其前后左右 10 公分处标定工位大小，四个角分别突出约 1 米的距离。

图 4.70　手动标定流程

（效果如图示 4.70）合理摆放四块标定布，并顺时针方向记录 0-7 号共八个角点的 X、Y、Z 世界坐标及对应摄像头相对 0 号点的 X、Y、Z 世界坐标。

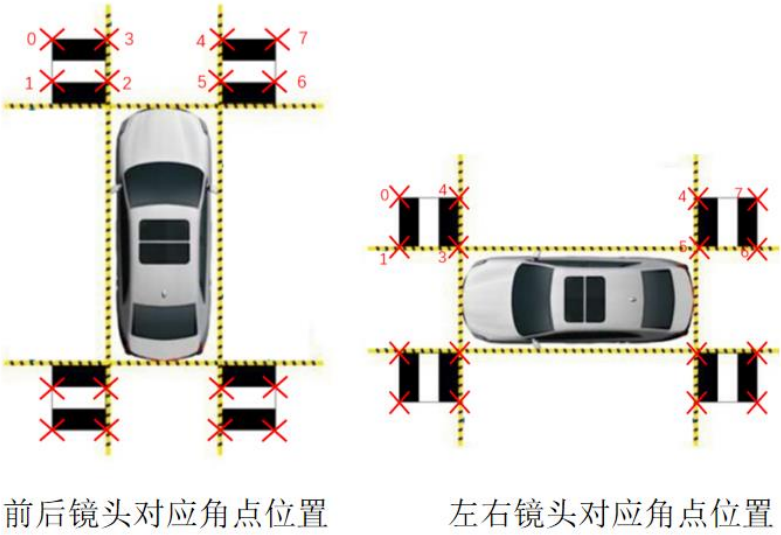


图 4.71 手动标定流程

（效果如图示 4.71）正确选择对应角点位置，角点位置选择越精确，360 环视全景拼接画面效果越好。

图 4.72　手动标定流程

（效果如图示 4.72）按照操作指引，从当前图像依次对每个镜头进行标定，同时记录各角点对应的 X,Y 图像坐标，最终四路图像选点标定完成后，保存并返回，进入高级设置，调整拼接融合区域。

图 4.73　手动标定流程

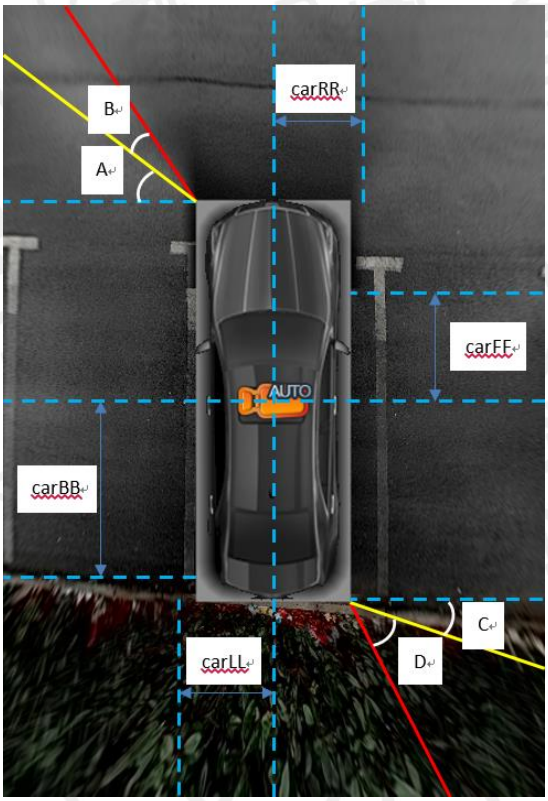


图 4.74　高级菜单设置说明

高级菜单设置说明（效果如图示 4.73、4.74）：

GL_3DModel_carLL：左车宽，数值越大左边黑底面积越大；

GL_3DModel_carRR：右车宽，数值越大右边黑底面积越大；

GL_3DModel_carFF：前车长，数值越大前边黑底面积越大；

GL_3DModel_carBB：后车长，数值越大后边黑底面积越大；

GL_3DModel_Front_Angle(A)：前融合起始角度；

GL_3DModel_Front_Range(B)：前融合区域角度；

GL_3DModel_Rear_Angle(C)：后融合起始角度；

GL_3DModel_Rear_Range(D)：后融合区域角度。

图 4.75　标定成功

根据实际测量车体上对应长宽，填写对应融合区域参数，并根据经验调整最终拼接融合效果，实际拼接效果根据拼接图像效果是否达到无缝拼接、是否断线、是否覆盖有效盲区、画面是否正常显示作为最终评分标准（效果如图示 4.75）。

五、智能网联汽车综合道路测试

（一）地图数据采集

1.录制地图准备

进行地图采集有两个先决条件：GPS 接收机已经被正确配置&标定、GPS 接收机上电后工作状态正常且 RTK 能够进入固定解。

录制地图准备

1.把自动驾驶车辆行驶到没有遮挡的空旷区域，在给接收机通电，并确定接收机已经通过串口与计算平台通信；

2.在计算平台终端，通过 cd 命令进入到 apollo 源码文件所在的目录；

3.执行./docker/scripts/dev_into.sh 进入 docker 容器；

4.进入容器后在 apollo 目录下了，执行./scripts/bootstrap.sh（启动 monitor 模块和 Dreamview 模块）；

5.在终端机中打开浏览器，输入 http://192.168.1.9:8888 或 localhost:8888，打开 DreamView，在 DreamView 中选择开启 GPS 功能模块，硬件中 GPS 显示为绿色，表明接收机接入正常；

6.在计算单元终端使用如下命令查看输出：rostopic echo /apollo/sensor/gnss/best_pose 如果有主题数据输出且为固定解状态（NARROW_INT），证明接收机正常工作（效果如图示 5.1）。

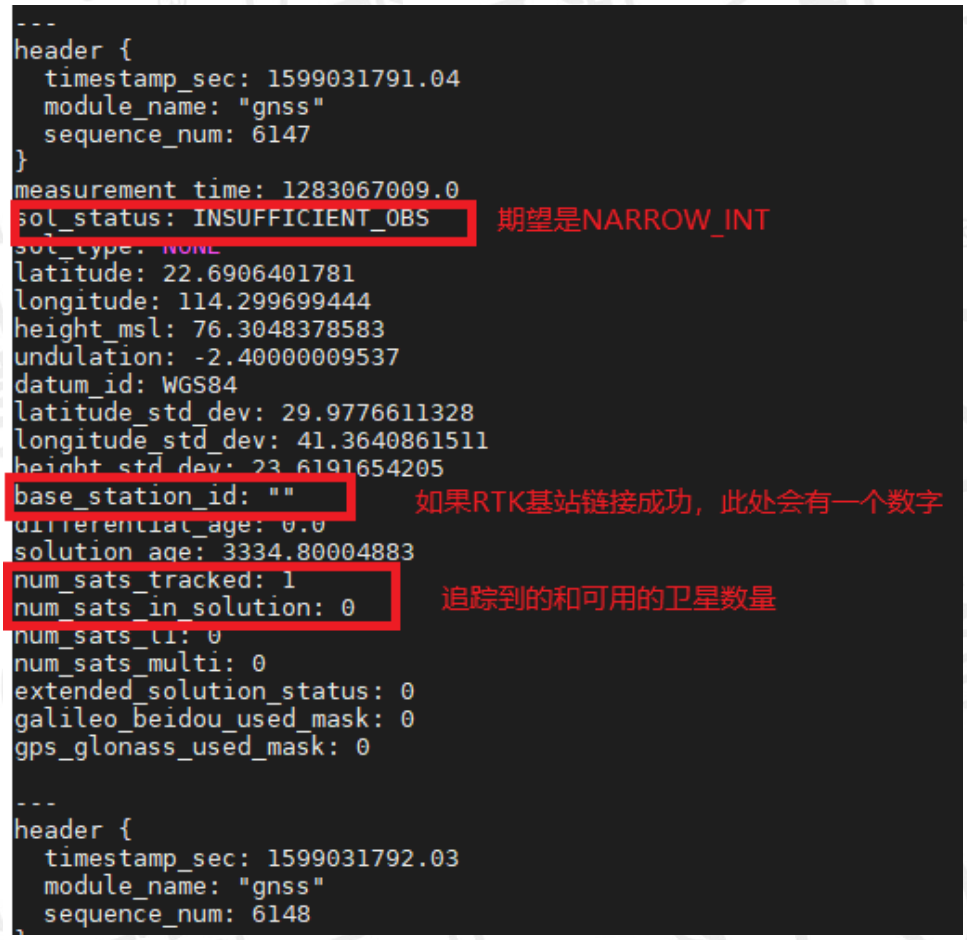


图 5.1　GPS 工作状态及 RTK 信息参数

2.录制高精度地图

在准备工作完成后，就可以进行地图数据的采集。

1.在 Dreamview 界面上依次启动 GPS、IMU、Localization 模块；等待 1~2 分钟，GPS、IMU 进入固定解状态（效果如图示 5.2）。

图 5.2　录制地图模块

2.将底盘遥控到测试路段起点，确保 GPS 定位状态稳定；点击主界面右侧的 Start Record Map 按钮开始录制地图。此时按钮转变为停止录制 Stop Record Map 遥控车辆按测试路段的中心线行驶（中途不要出现倒车）；地图录制过程中每记录 1000 个 GPS 坐标点，会在 Console

中显示相应的提示信息（效果如图示 5.3）。

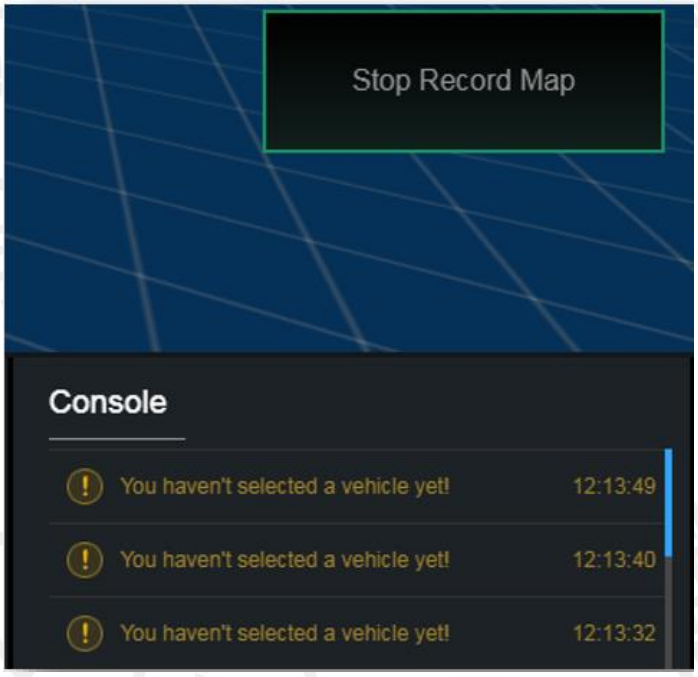


图 5.3　对应模块提示信息

3.到达路段终点时，点击停止录制按钮，稍等一段时间 Console 中会打印出地图制作成功的信息（包含生成的地图名称，以录制的开始时间为名称）；然后可以在地图下拉列表中看到新录制的地图,例如 2020-09-24-14-37-26(地图名，根据时间自动生成)（效果如图示 5.4）。

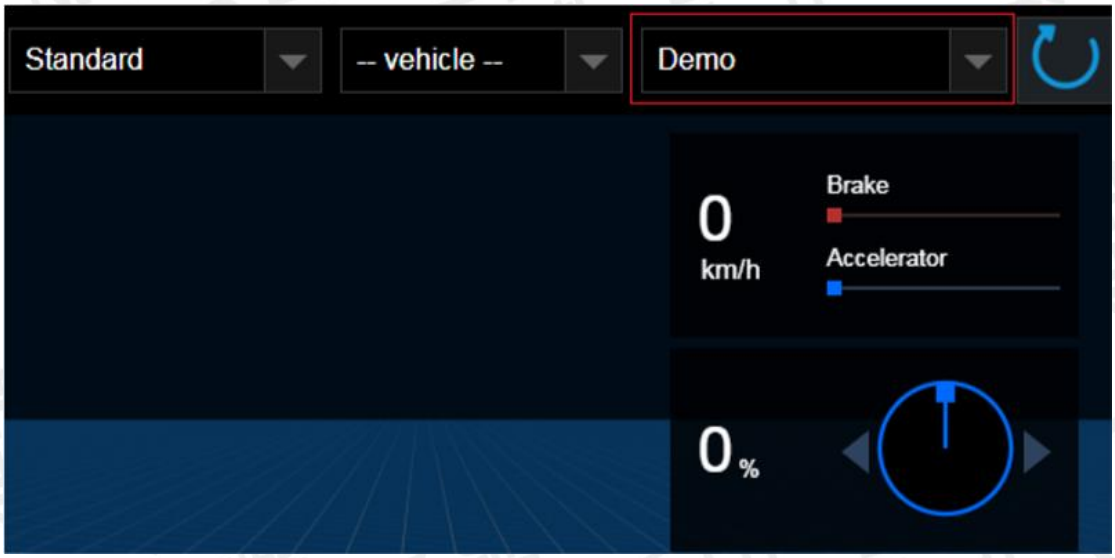


图 5.4　地图录制 Demo

（二）添加红绿灯

1.前提条件

1.已经在 Dreamview 界面上通过 Start Record Map 成功录制一个地图，并在 /apollo/modules/map/data 目录下存在相应的文件夹，此处假设地图文件名称为：2020-09-24-14-37-26

2. 删掉 /apollo/modules/map/data/2020-09-24-14-37-26 目录下的 sim_map.bin、routing_map.bin，删掉之后在该目录下只存在以下文件（效果如图示 5.5）。

2020-09-24-14-37-26.txt		base_map.txt
default_end_way_point.txt	routing_map.txt	sim_map.txt

图 5.5 map 文件夹

2.添加步骤

1.打开/apollo/modules/map/data/2020-09-24-14-37-26/base_map.txt 文件，在文件末尾添加 2 个新元素：overlap 和 signal（效果如图示 5.6）。

```
overlap {
  id {
    id: "overlap_4817"
  }
  object {
    id {
      id: "1"
    }
    lane_overlap_info {
      start_s: 25.499341
      end_s: 25.599341
      is_merge: false
    }
  }
```

```
    object {
      id {
        id: "2505"
      }
      signal_overlap_info {
      }
    }
  }
  signal {
    id {
      id: "2505"
    }
    boundary {
      point {
        x: 231144.25606501789
        y: 2515176.7369439765
        z: 39.29290285
      }
      point {
        x: 231144.25606501789
        y: 2515176.2369439765
        z: 39.29290285
      }
      point {
        x: 231143.75606501789
        y: 2515176.2369439765
        z: 39.79290285
      }
```

```
    point {
      x: 231143.75606501789
      y: 2515176.7369439765
      z: 39.79290285
    }
  }
  subsignal {
    id {
      id: "0"
    }
    type: CIRCLE
    location {
      x: 231144.25606501789
      y: 2515176.7369439765
      z: 40.29290285
    }
  }
  subsignal {
    id {
      id: "1"
    }
    type: CIRCLE
    location {
      x: 231144.25606501789
      y: 2515176.7369439765
      z: 40.39290285
    }
  }
```

```
subsignal {
  id {
    id: "2"
  }
  type: CIRCLE
  location {
    x: 231144.25606501789
    y: 2515176.7369439765
    z: 40.49290285
  }
}
overlap_id {
  id: "overlap_4817"
}
type: MIX_3_VERTICAL
stop_line {
  segment {
    line_segment {
      point {
        x: 231144.2092253688
        y: 2515178.4873587429
      }
      point {
        x: 231145.63070861
        y: 2515178.628037379
      }
      point {
        x: 231147.03385822
```

```
                y: 2515178.62491749
            }
          }
        }
      }
    }
```

图 5.6　添加文件

Overlap 和 Signal 元素中的 id.id（见表格中标红的文字）用于唯一标识一个元素，在地图文件中不能重复；Overlap 和 Signal 元素相互引用，Overlap 中的 object.id.id 用于关联红绿灯元素，此处填写红绿灯的 id.id 字段的值”2505”，Signal 中的 overlap_id.id 用于标识关联的 Overlap 的 id.id 字段的值。

Signal 元素中描述了信号灯的形状（type: MIX_3_VERTICAL）和红绿灯上灯的个数（subsignal 子元素的数量），根据 type 字段指定的类型来决定添加几个 subsignal。Signal 元素中的 stop_line 用于在车道上标出停止线，这次都是以绝对坐标来标明的，后续会说到如何确定坐标的值。

2.在上面的文件中找到 lane 元素，添加 overlap_id 子元素，此处里面的 id 是上面新添加的 Overlap 元素的 id.id（效果如图示 5.7）。

```
lane {
  id {
    id: "1"
  }
  central_curve {…
  }
  left_boundary {…
  }
  right_boundary {…
  }
  overlap_id {
    id: "overlap_4817"
  }
  length: 100
  speed_limit: 1.5
  type: CITY_DRIVING
  turn: NO_TURN
  direction: BIDIRECTION
}
```

图 5.7　添加 overlap_id

3.以上只是把元素添加到 base_map.txt 文件中，元素中还有很多地方需要填写绝对坐标，此时可以打开 Dreamview，然后在地图下拉列表中选择 2020-09-24-14-37-26 这个地图，然后点击左侧 Route Editing 按钮（默认启动时这个按钮时灰色的，不能点击，这时可以打开 SimControl 开关）（效果如图示 5.8）

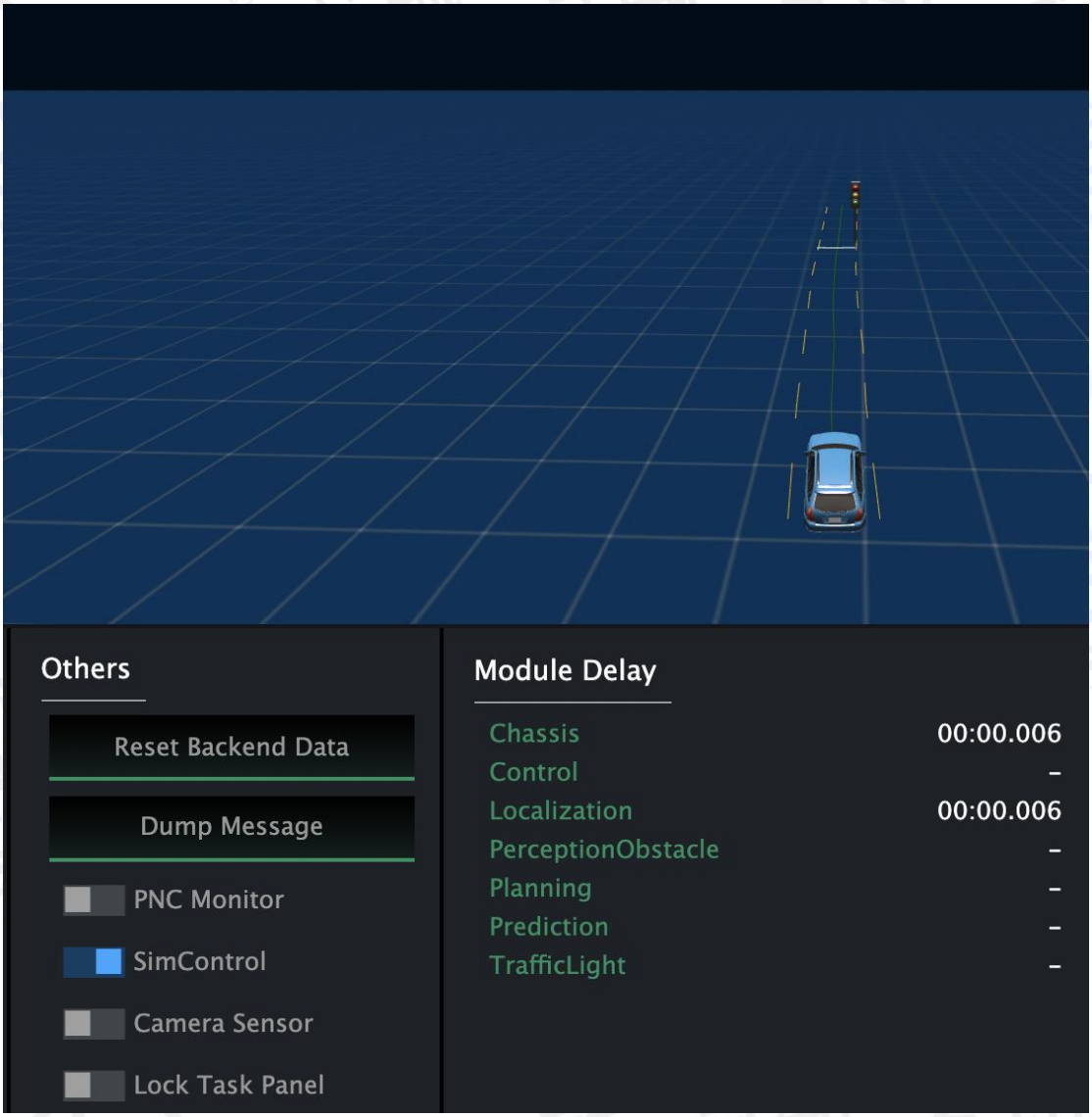


图 5.8　打开地图点击 SimControl

4.点击右侧 Route Editing 按钮（效果如图示 5.9）。

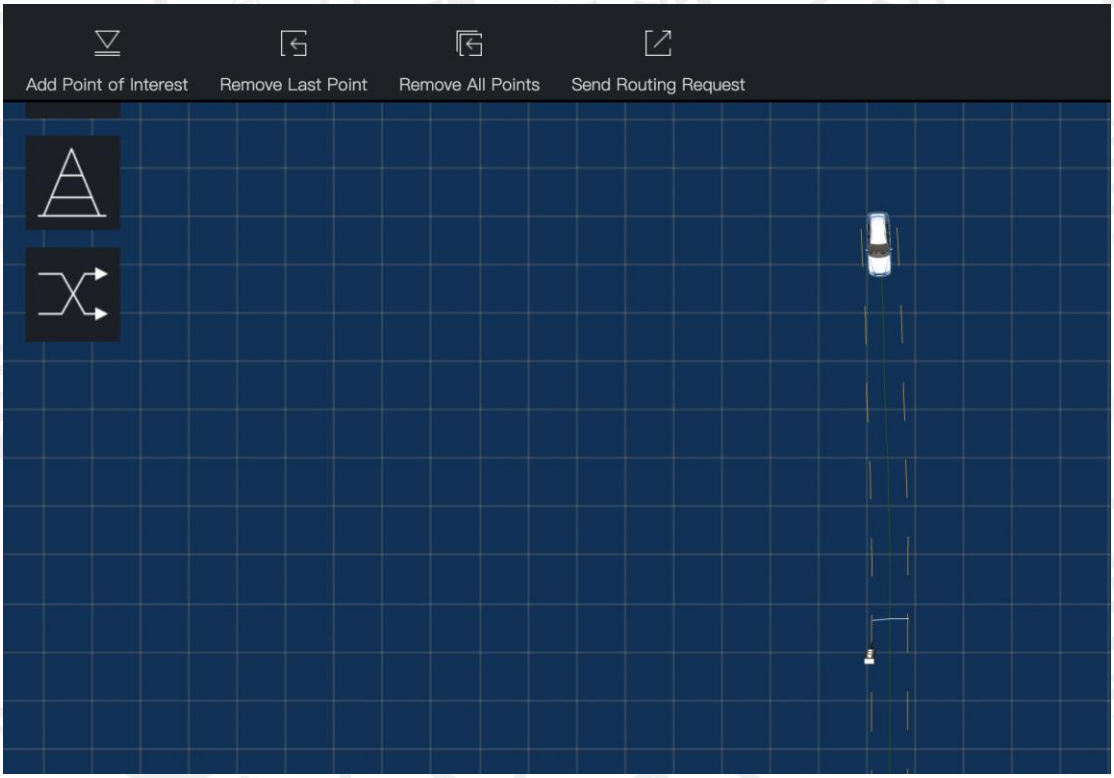


图 5.9　点击 Route Editing 按钮显示地图

5.鼠标在地图上移动的过程中界面右下角会显示鼠标所在位置在地图上的绝对坐标（x、y）值（效果如图示 5.10）。

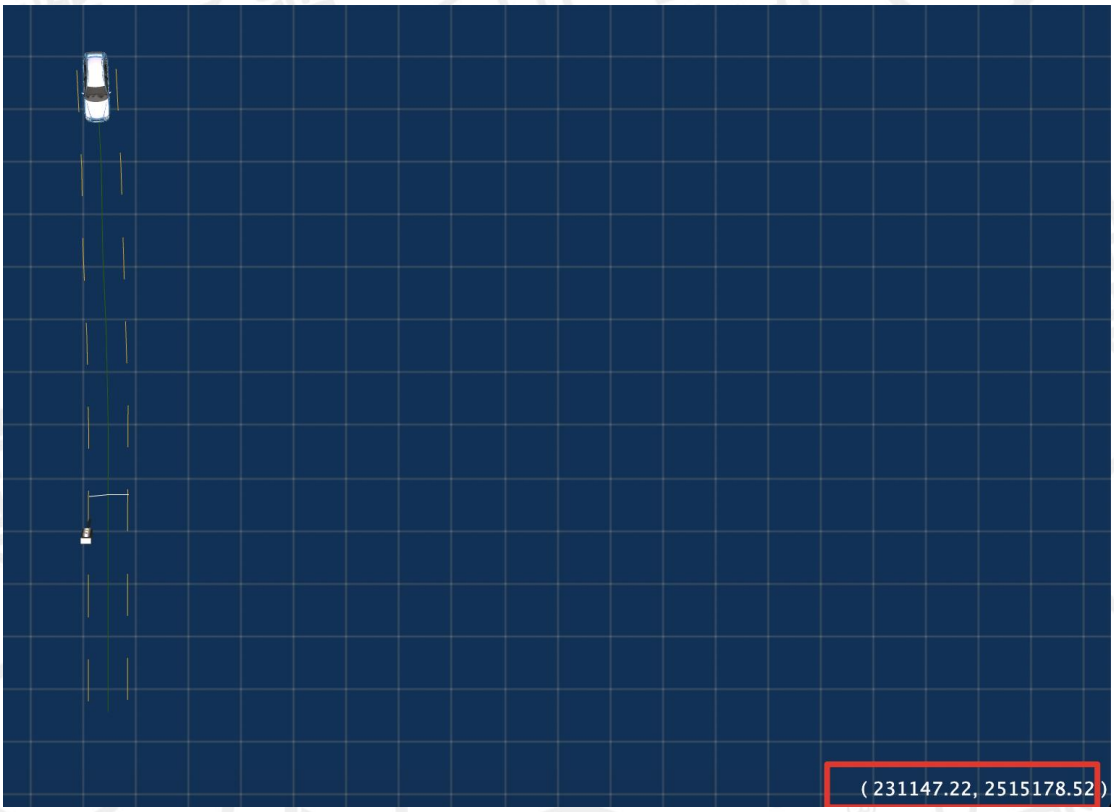


图 5.10　确定绝对坐标

6.根据红绿灯所在的位置在地图上截取相应的 x、y 坐标，填写到 Signal 元素中。

7.更改 base_map 文件（效果如图示 5.11）。

(1).lane1_-1 ,only make sure that points of central_curve are correct,and central_curve. right_boundary left_boundary length keep be same（更改长度 length，单位：m）

(add “ successor_id {

id: "1_-2"

}

overlap_id {

id: "overlap_4817"

}”)

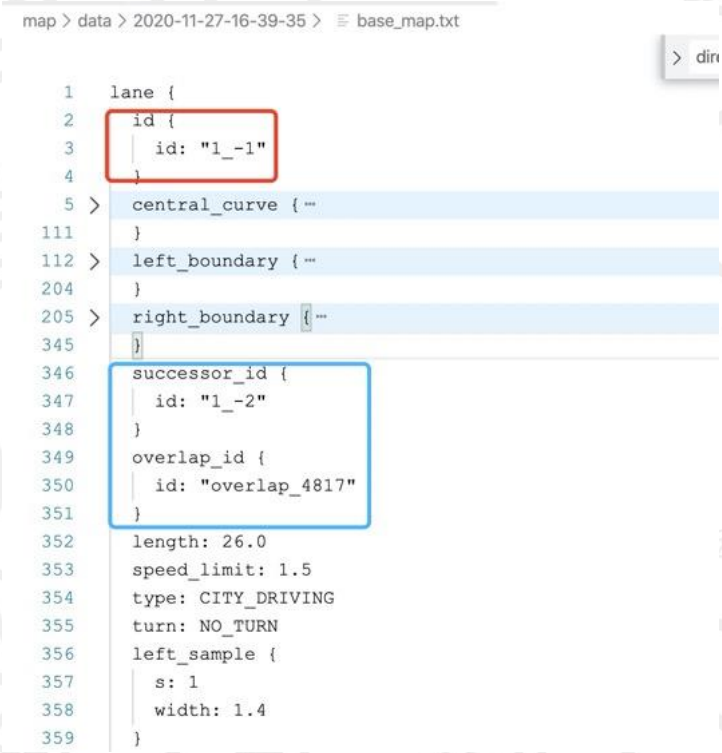


图 5.11 更改第一段文件

(2)更改 lane 文件（效果如图示 5.12）。

(before road,behind lane)

(add” predecessor_id {

id: "1_-1"

} ”)

map > data > 2020-11-27-16-39-35 > base_map.txt

```
1    > lane {⋯
564    }
565    lane {
566      id {
567        id: "1_-2"
568      }
569    > central_curve {⋯
771      }
772    > left_boundary {⋯
1000     }
1001   > right_boundary {⋯
1181     }
1182     predecessor_id {
1183       id: "1_-1"
1184     }
1185     length: 100.0
1186     speed_limit: 1.5
1187     type: CITY_DRIVING
1188     turn: NO_TURN
1189     left_sample {
1190       s: 1
1191       width: 1.4
1192     }
1193     left_sample {
1194       s: 2
1195       width: 1.4
1196     }
```

图 5.12　更改第二段文件

(3)更改 road 文件（效果如图示 5.13）。

(add” lane_id {

id: "1_-1"

}

lane_id {

id: "1_-2"

}”)

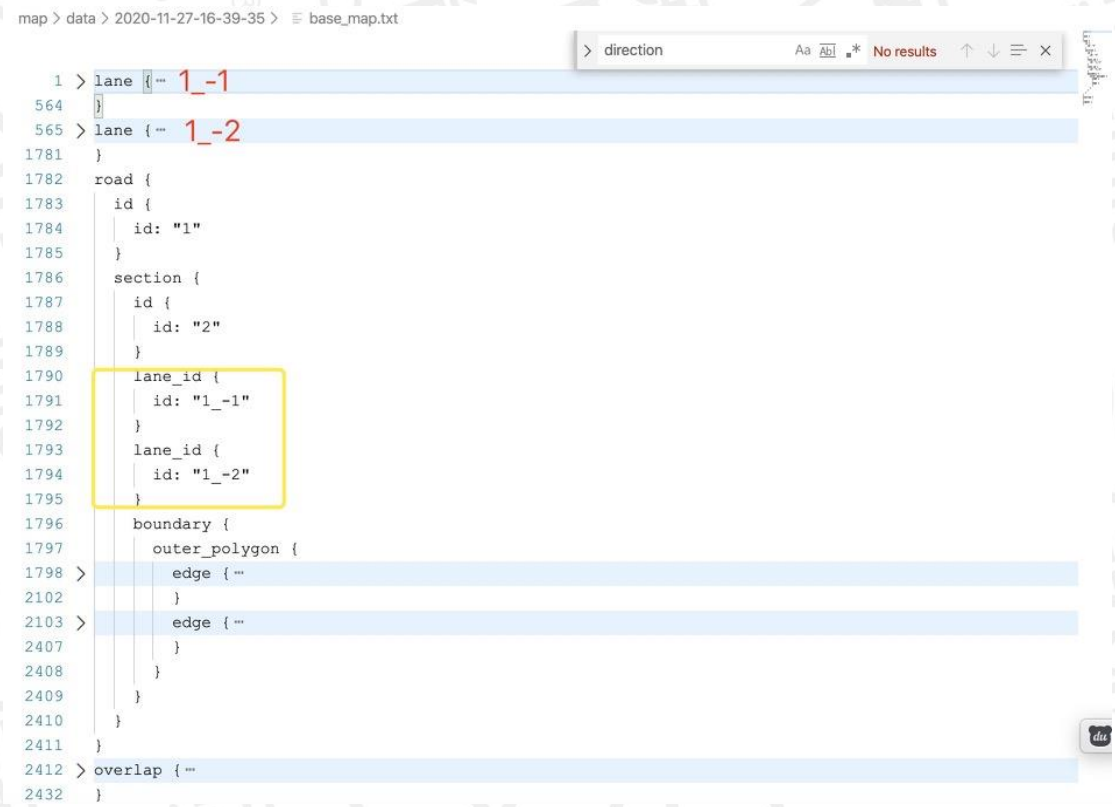


图 5.13　将第一段路线和第二段路线改相连

8.更新 map，进入 docker 环境，然后创建两个文件（效果如图示 5.14）。

/apollo/bazel-bin/modules/map/tools/sim_map_generator

--map_dir=/apollo/modules/map/data/2020-11-27-16-39-35

-output_dir=/apollo/modules/map/data/2020-11-27-16-39-35

/apollo/bazel-bin/modules/routing/topo_creator/topo_creator

-flagfile=/apollo/modules/routing/conf/routing.conf

-map_dir=/apollo/modules/map/data/2020-11-27-16-39-35

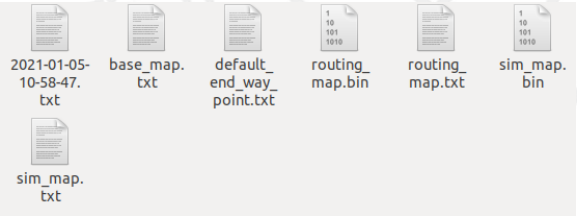


图 5.14　添加文件

9.在 Docker 容器中执行./scripts/bootstrap.sh stop 关闭 Dreamview，然后再执行./scripts/bootstrap.sh 启动 Dreamview，然程序重新加载地图；在浏览器中访问 Dreamview，

打开 SimControl 按钮，查看红绿灯是否加到正确的位置（效果如图示 5.15）。

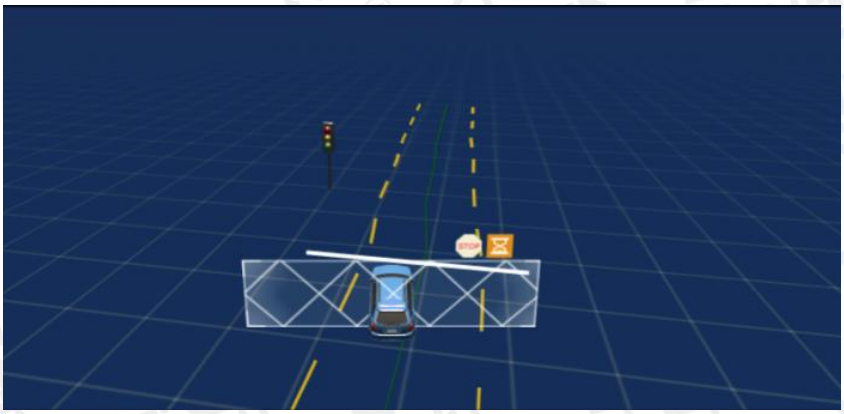


图 5.15　红绿灯添加图

（三）自动驾驶功能测试

1.循迹自动驾驶功能测试

1.将车辆行驶到测试路段的起点，确保车辆完成在车道线内（效果如图示 5.16）。

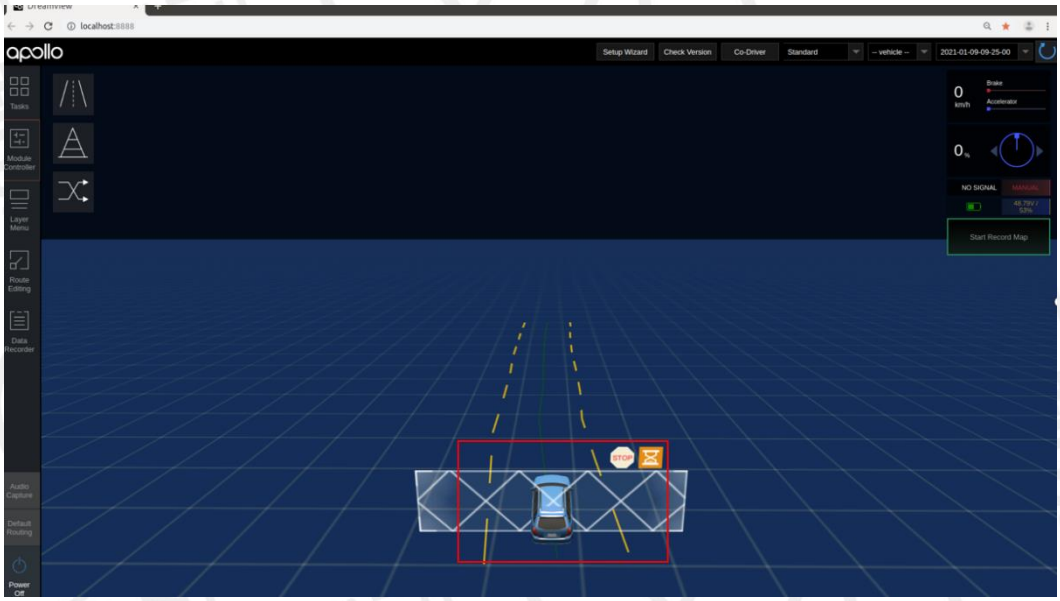


图 5.16　车辆行驶至测试路段

2.将右上角 Demo 改成录制的高精度地图，例如 2021-01-09-09-25-00 地图（效果如图示 5.17）。

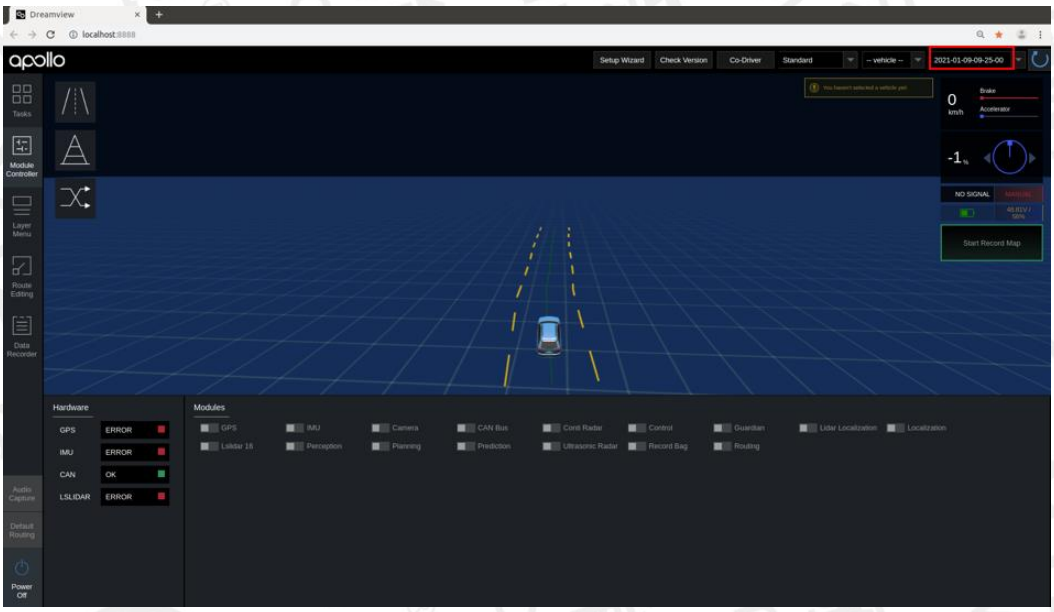


图 5.17　更改地图

3.在 Dreamview 界面上依次启动 GPS、IMU、CanBus、Guardian、Localization、Planning、Routing 模块。等待 1~2 分钟，GPS、IMU 进入固定解状态（效果如图示 5.18）。

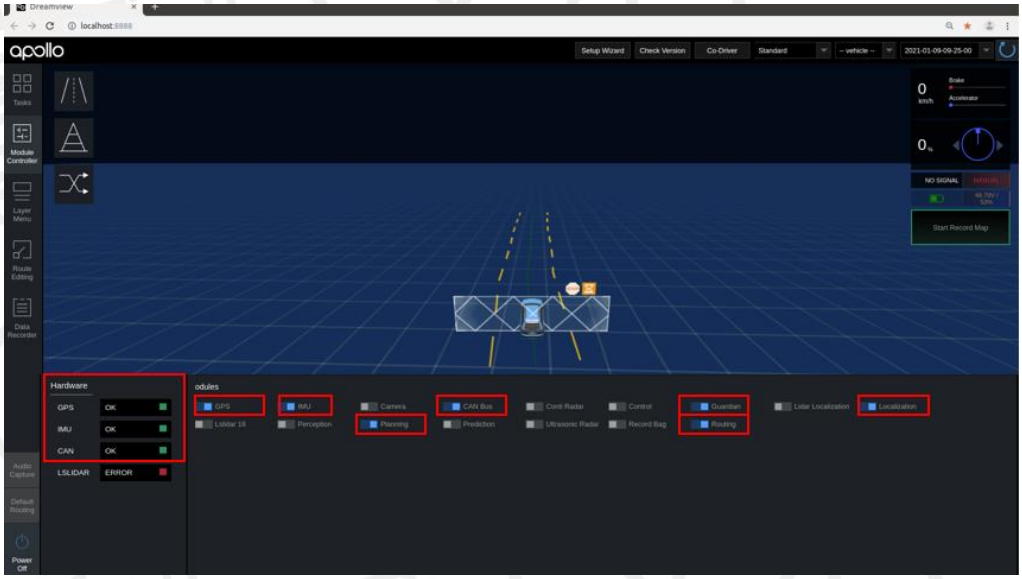


图 5.18　打开 Dreamview 界面内模块

4.进行路线规划，选择车辆停止线位置，选择车辆停止点时选择靠近车道中心位置（效果如图示 5.19）。

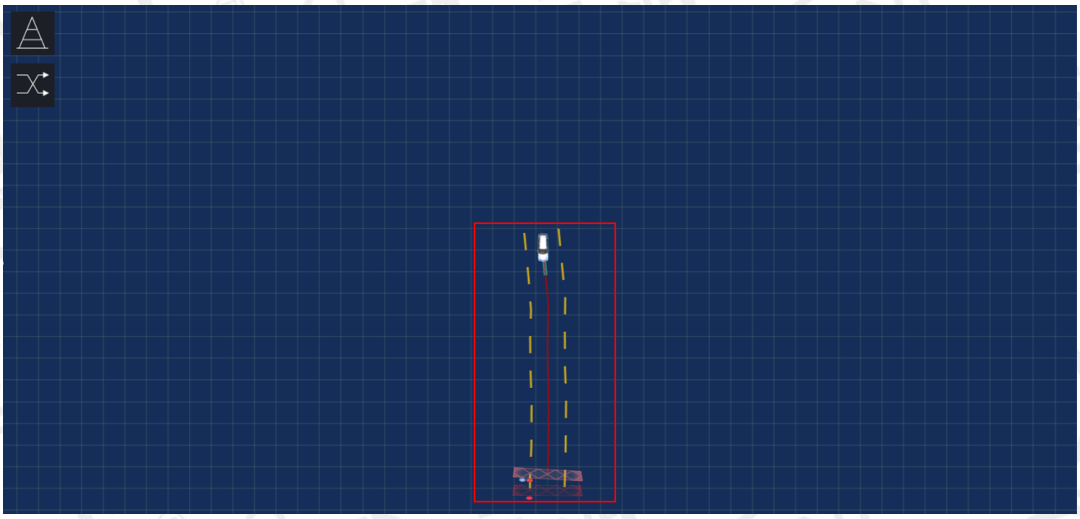


图 5.19　规划路线

5.返回 Dreamview 界面启动 Control，Console 中显示相应的提示信息（效果如图示 5.20、5.21）Module Delay 中显示相应的提示信息（效果如图示 5.22）。

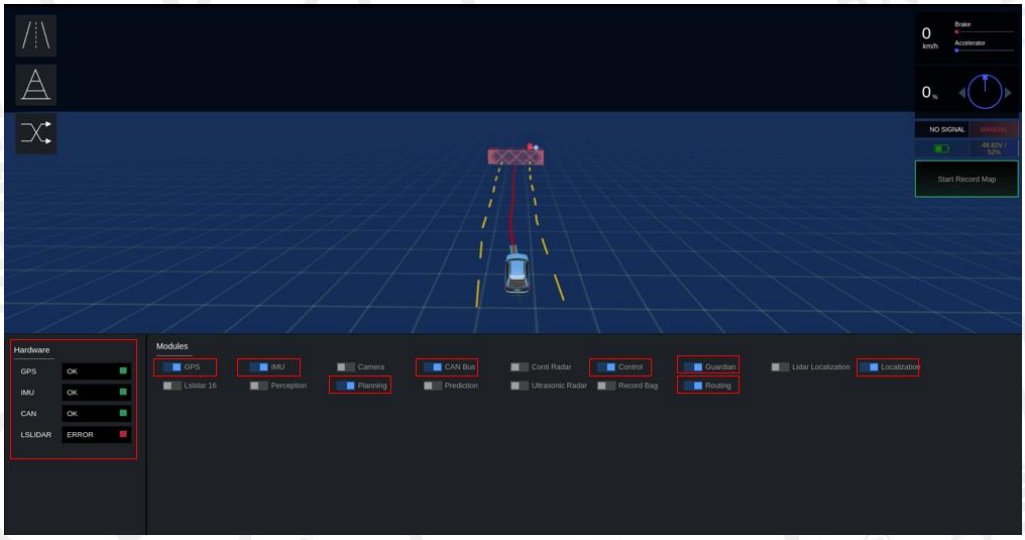


图 5.20　启动 Control

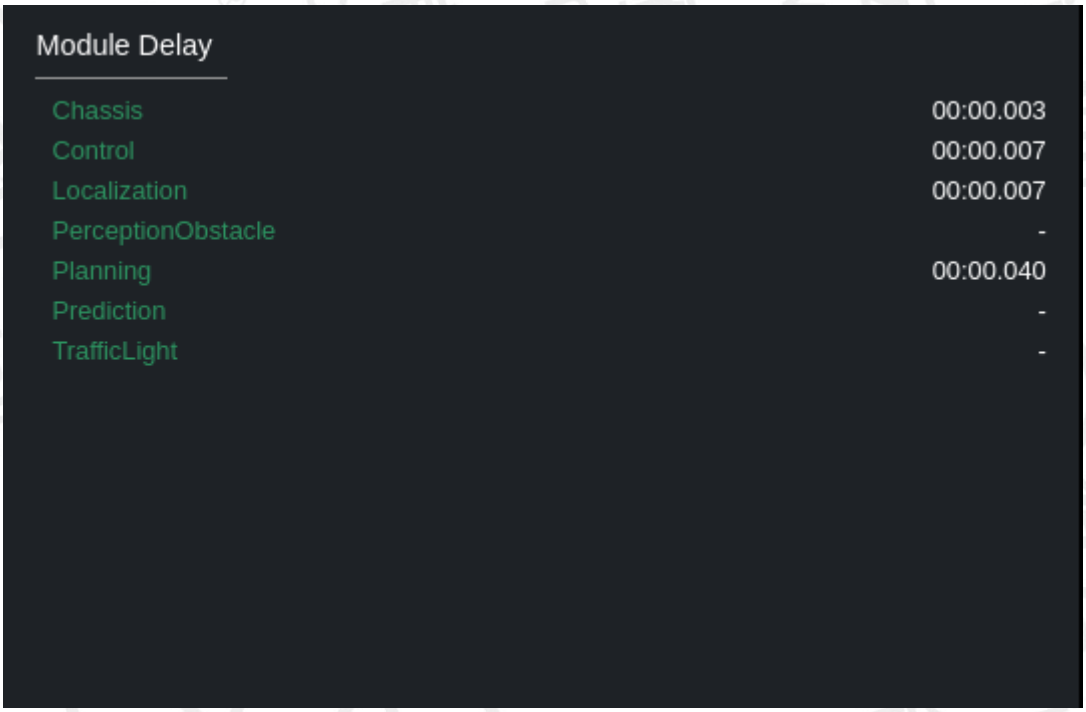


图 5.21　Module Delay 提示信息

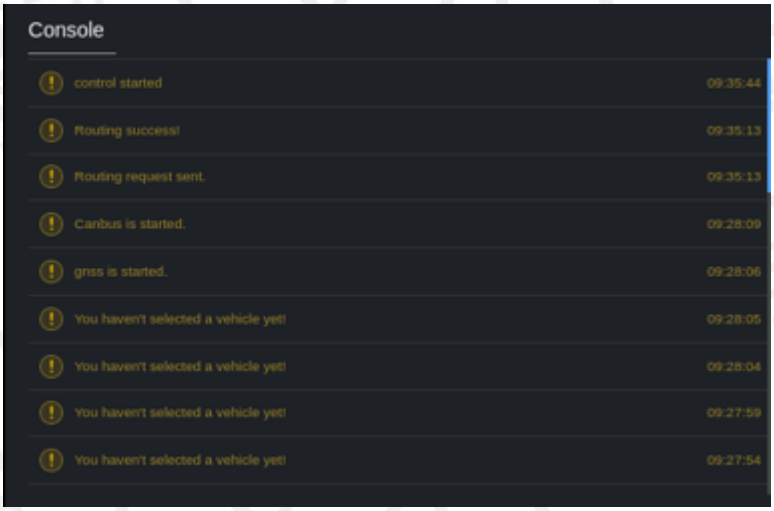


图 5.22　Console 提示信息

6.将遥控器切换到自动挡，并且点击 Start Auto,启动自动驾驶循迹（效果如图示 5.23）。

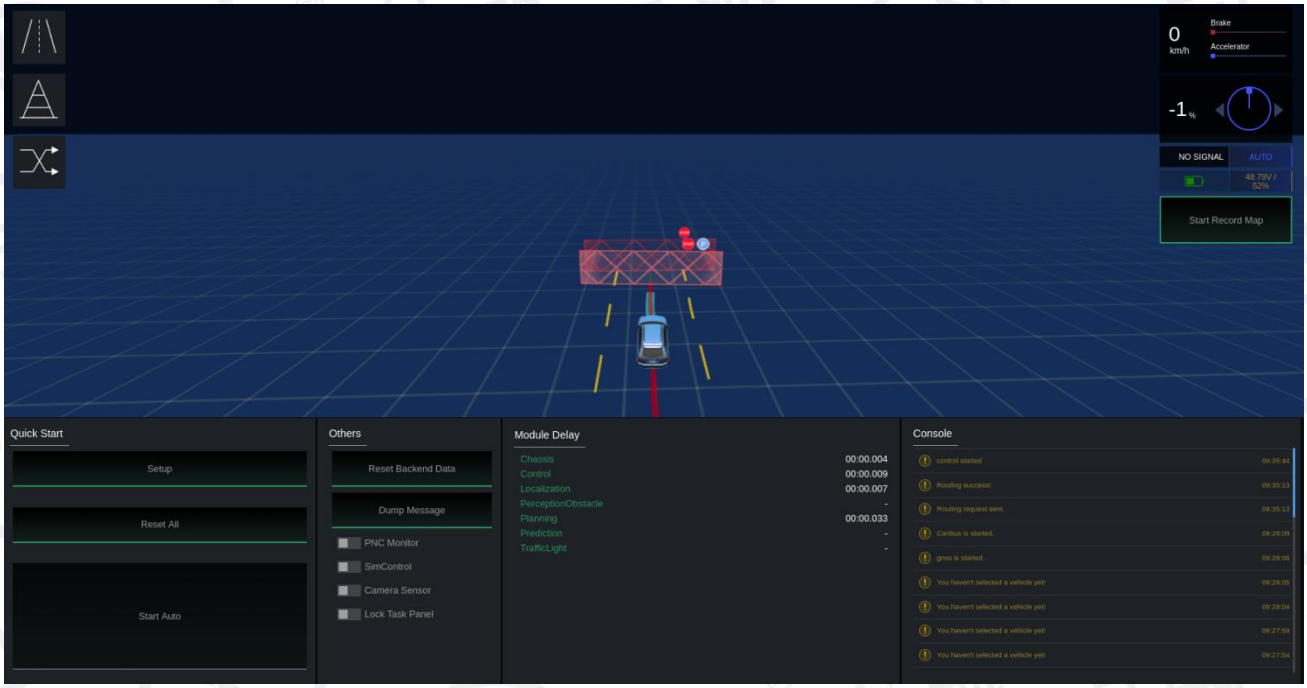


图 5.23　自动驾驶循迹

2.停障自动驾驶功能测试

（1）使用超声波雷达实现自动驾驶功能停障的测试

1.将车辆行驶到测试路段的起点。

2.将右上角 Demo 改成录制的高精度地图。

3.在 Dreamview 界面上依次启动 GPS、IMU、CanBus、Guardian、Localization、Planning、Routing、ultransonic Radar 模块。等待 1~2 分钟，GPS、iMU 进入固定解状态（效果如图示 5.24）。

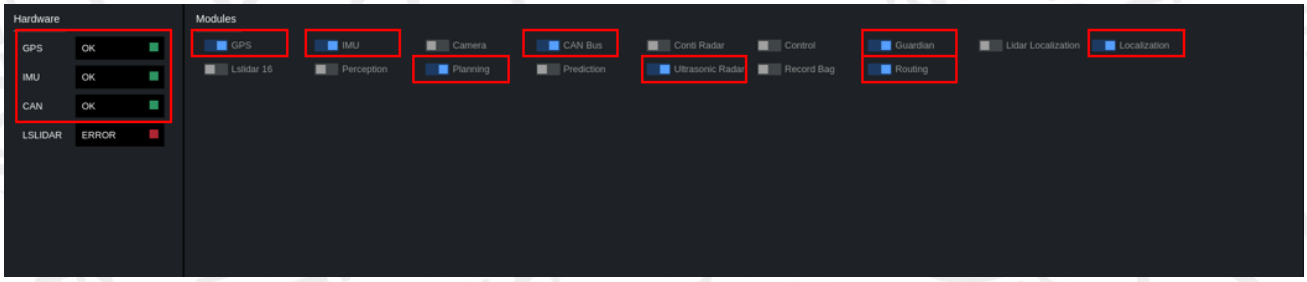


图 5.24　打开 Dreamview 界面内模块

4.进行路线规划，选择车辆停止线位置（效果如图示 5.25）。

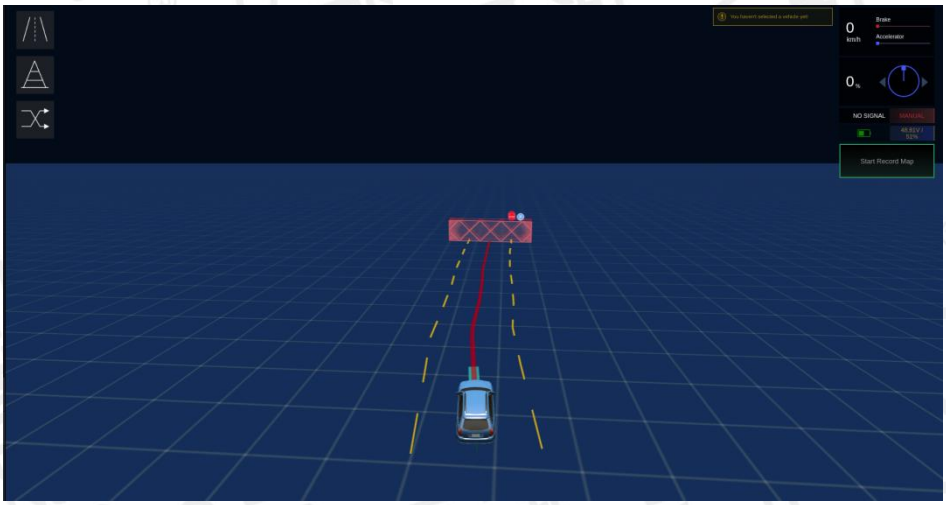


图 5.25　路线规划

5.返回 Dreamview 界面启动 Control，Console 中显示相应的提示信息，如图 5.27 所示，Module Delay 中显示相应的提示信息，如图 5.28 所示。

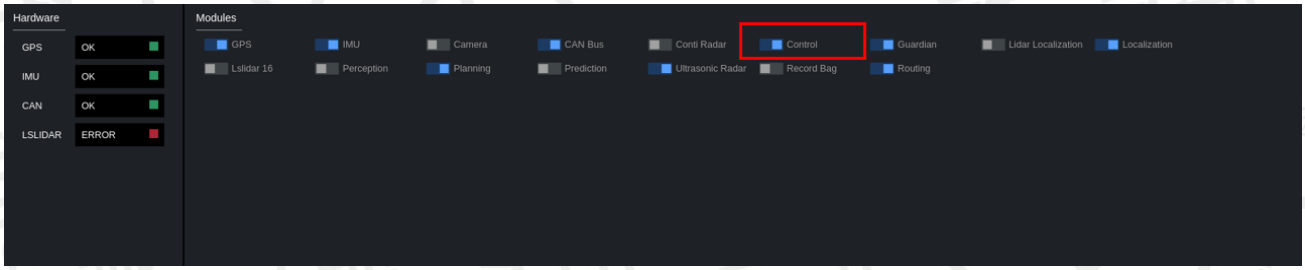


图 5.26 启动 Control 模块

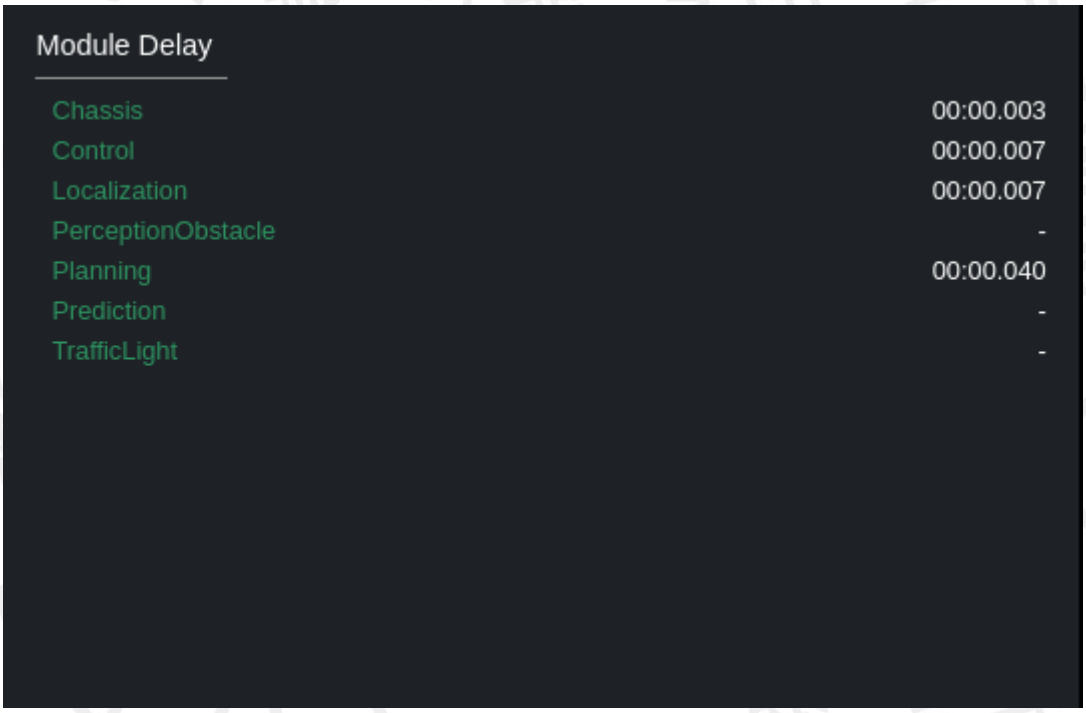


图 5.27　Module Delay 提示信息

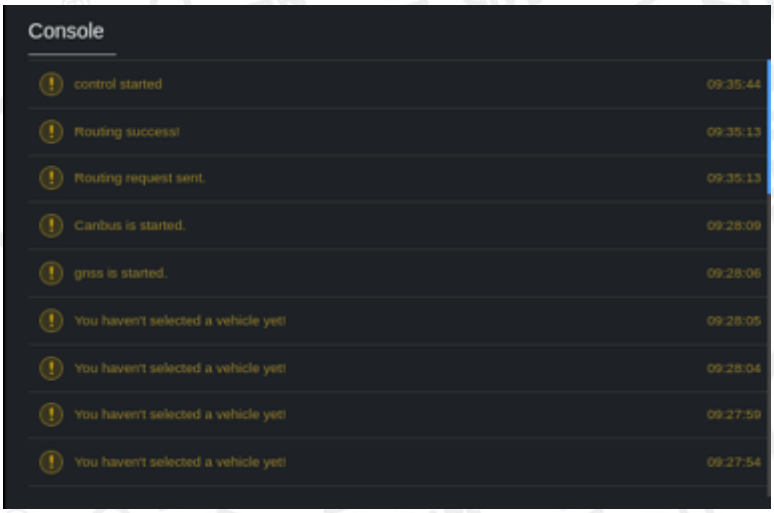


图 5.28　Console 提示信息

6.将遥控器切换到自动挡，并且点击 Start Auto，启动自动驾驶（效果如图示 5.29）。

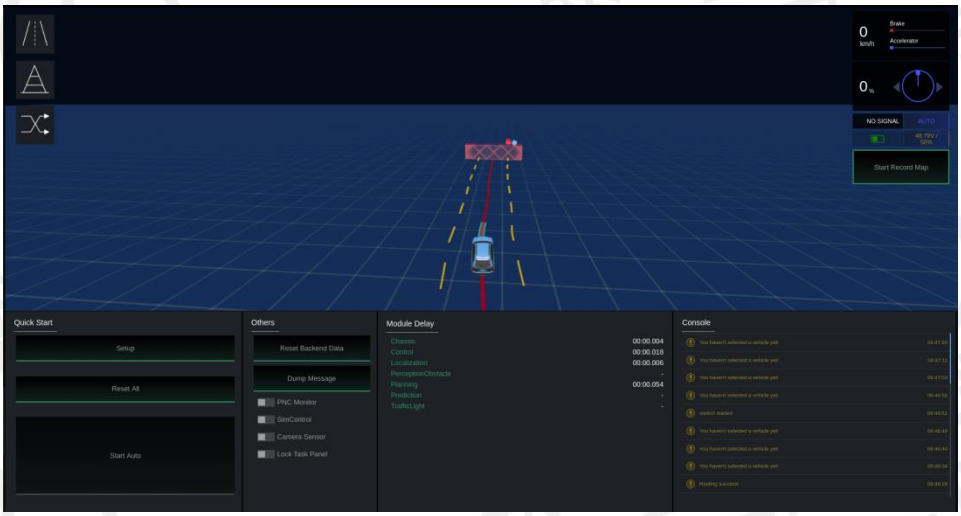


图 5.29　自动驾驶

7.在行驶过程中碰到障碍物完全挡住测试路段，车辆会停止在障碍物前（效果如图示 5.30），如障碍物移开，车辆继续行驶（效果如图示 5.31）。

图 5.30　停障功能测试

图 5.31　停障后移开障碍物继续行驶

（2）使用双目摄像头实现自动驾驶功能停障的测试

1.将车辆行驶到测试路段的起点。

2.将右上角 Demo 改成录制的高精度地图。

3.在 Dreamview 界面上依次启动 GPS、IMU、CanBus、Guardian、Localization、Planning、Routing、Camera、Perception 等模块（如下图）。等待 1~2 分钟，GPS、IMU 进入固定解状态（效果如图示 5.32）。

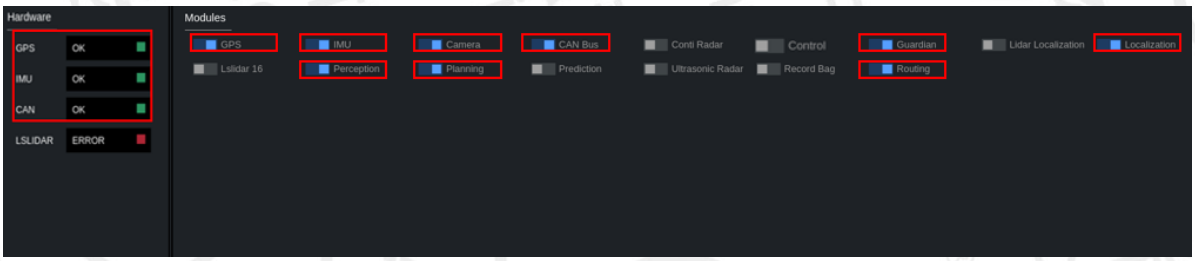


图 5.32　打开 Dreamview 界面内模块

4.移动车辆到起始线位置。

5.返回 Dreamview 界面启动 Control（效果如图示 5.33），Console 中显示相应的提示信息，（效果如图示 5.35），Module Delay 中显示相应的提示信息（效果如图示 5.34）。

图 5.33　启动 Control 模块

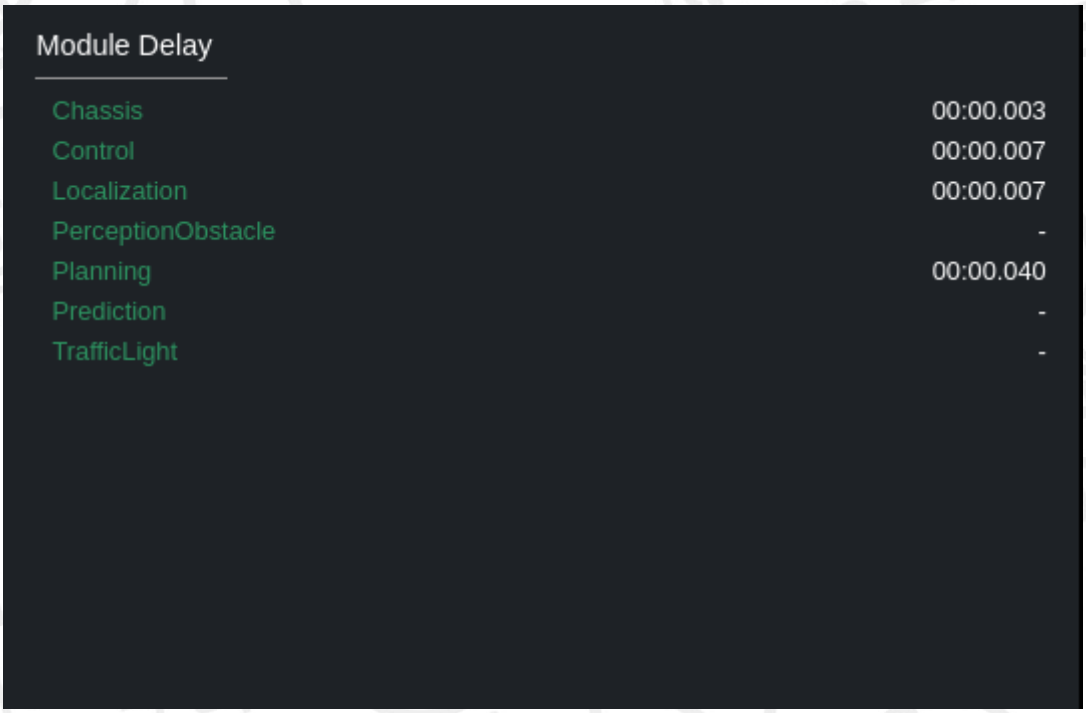


图 5.34　Module Delay 提示信息

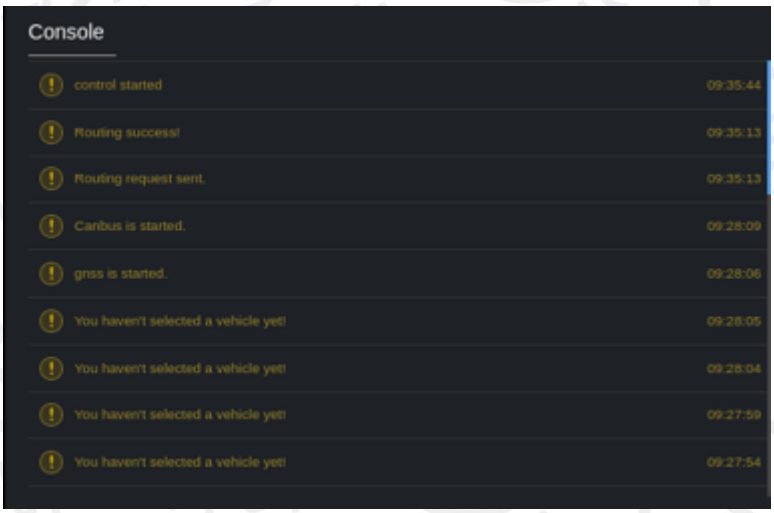


图 5.35　Console 提示信息

6.将遥控器切换到自动挡，并且点击 Start Auto,启动自动驾驶（效果如图示 5.36）。

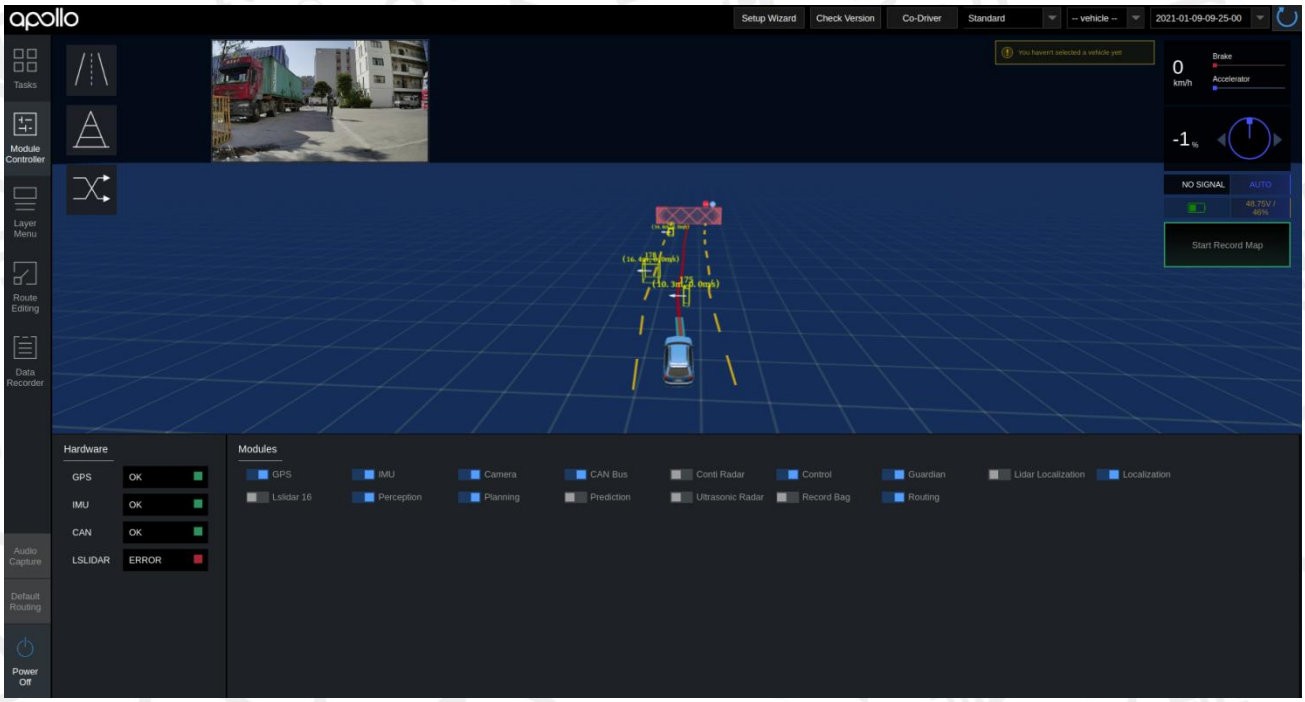


图 5.36 停障功能测试

可以看到在屏幕上显示障碍物与车辆位置关系。

7.在行驶过程中碰到障碍物完全挡住测试路段，车辆会停止在障碍物前（效果如图示 5.37、5.38）；如障碍物移开，车辆继续行驶（效果如图示 5.39）。

图 5.37 双目摄像头画面

图 5.38　停障功能测试

图 5.39　停障后移开障碍物

8.在行驶过程中碰到障碍物未完全挡住测试路段，车辆继续行驶（效果如图示 5.40、5.41、5.42）。

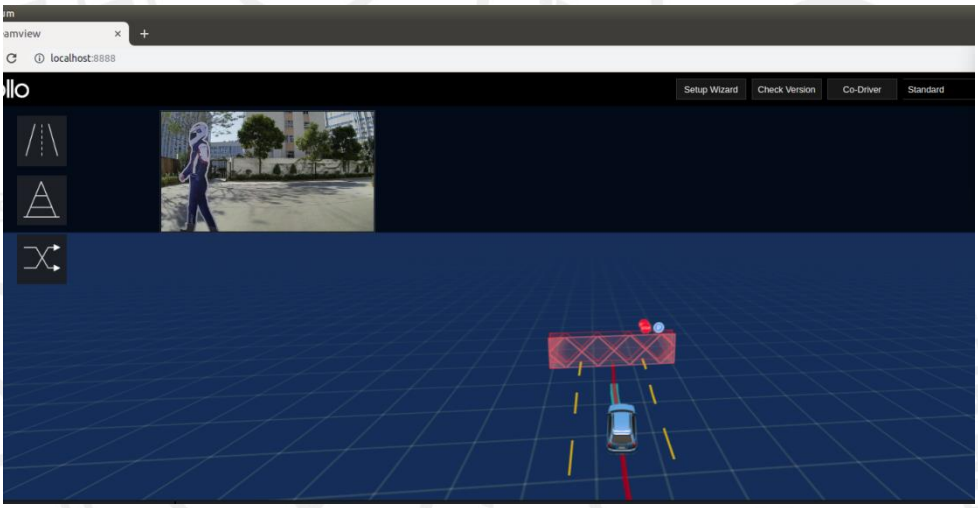


图 5.40 未完全挡住路段测试

图 5.41 自动驾驶功能测试

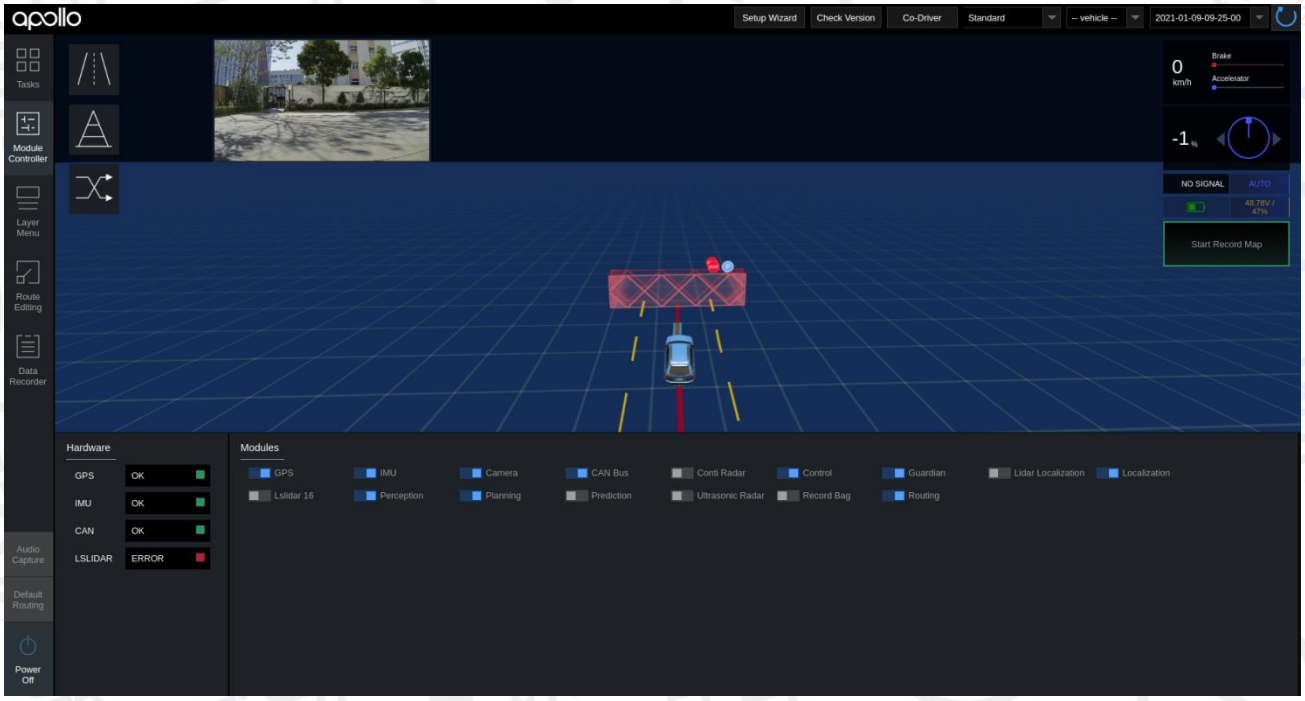


图 5.42 继续自动驾驶

3.避障自动驾驶功能测试

1.将车辆行驶到测试路段的起点。

2.将右上角 Demo 改成录制的高精度地图。

3.在 Dreamview 界面上依次启动 GPS、IMU、CanBus、Guardian、Localization、Planning、

Routing、Camera、Perception、Prediction、Ultransonic Radar 模块。等待 1~2 分钟，GPS、IMU 进入固定解状态（效果如图示 5.43）。

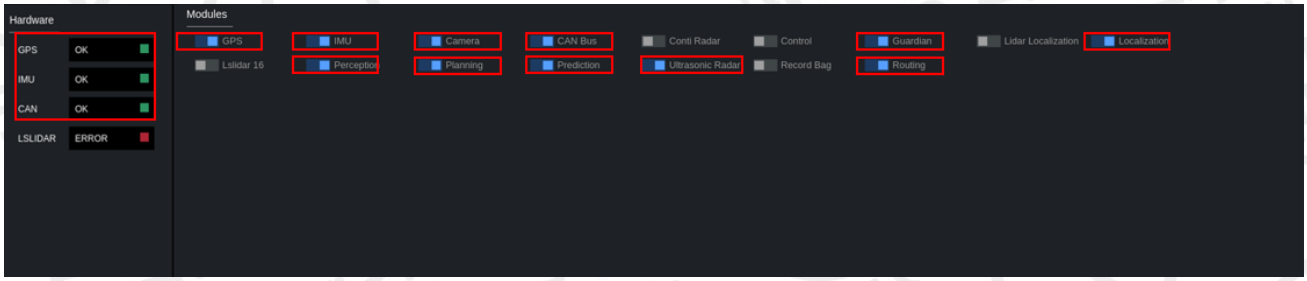


图 5.43　打开 Dreamview 界面内模块

4.进行路线规划，选择车辆停止线位置。

5.返回 Dreamview 界面启动 Control（效果如图示 5.44），Console 中显示相应的提示信息（效果如图示 5.46） Module Delay 中显示相应的提示信息（效果如图示 5.45）。

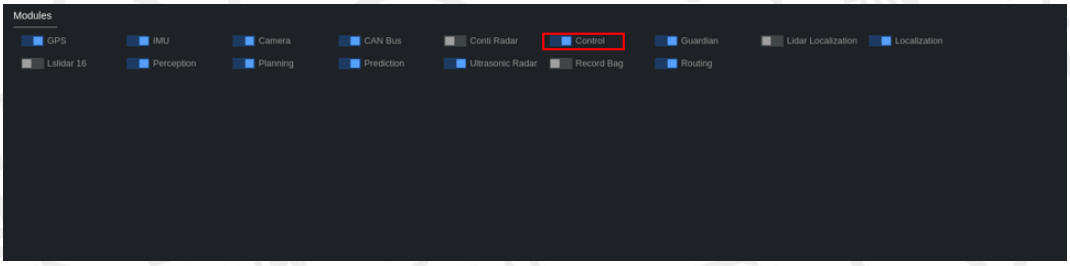


图 5.44　启动 Control 模块

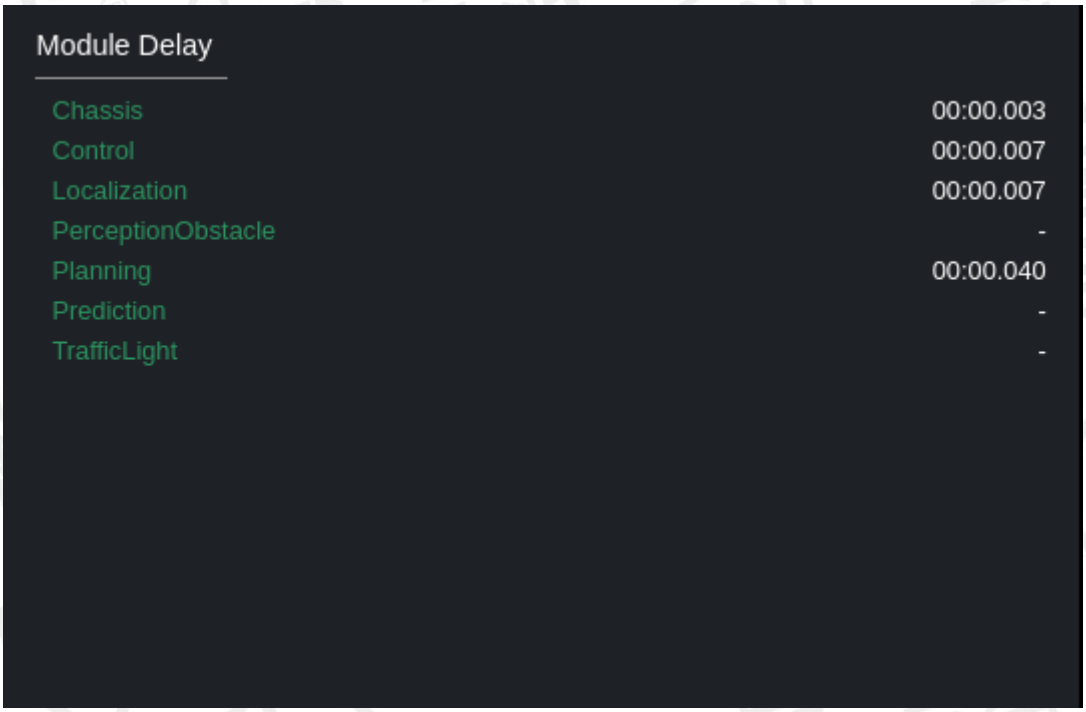


图 5.45　Module Delay 提示信息

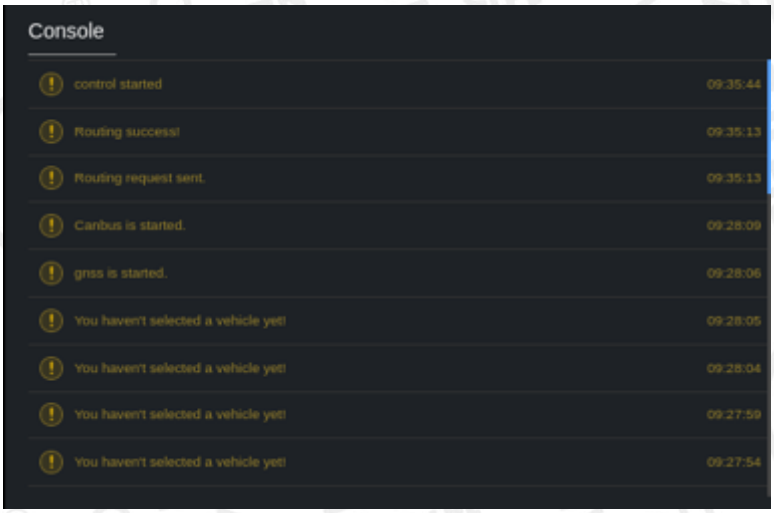


图 5.46　Console 提示信息

6.将遥控器切换到自动挡，并且点击 Start Auto,启动自动驾驶（效果如图示 5.47）。

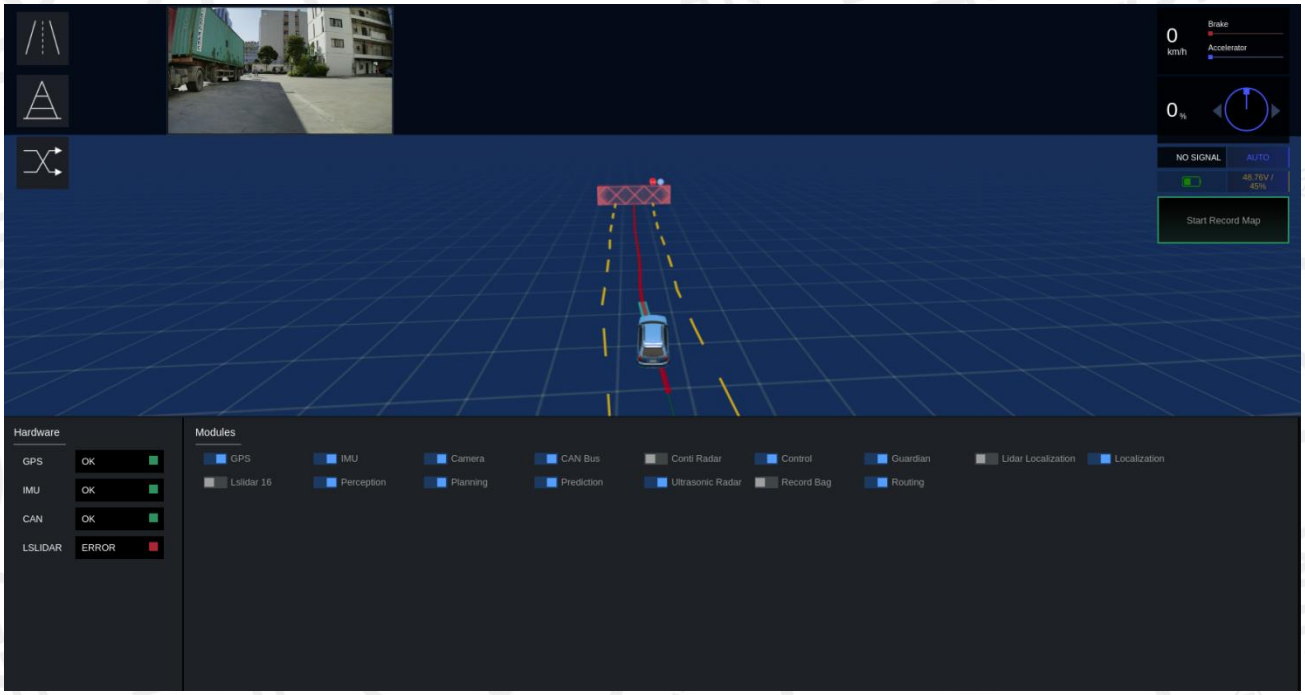


图 5.47　自动驾驶功能

7.在行驶过程中遇到障碍物，车辆会绕过障碍物（效果如图示 5.48、5.49、5.50）；并且回到在车辆原本规划路线内车继续行驶（效果如图示 5.51）。

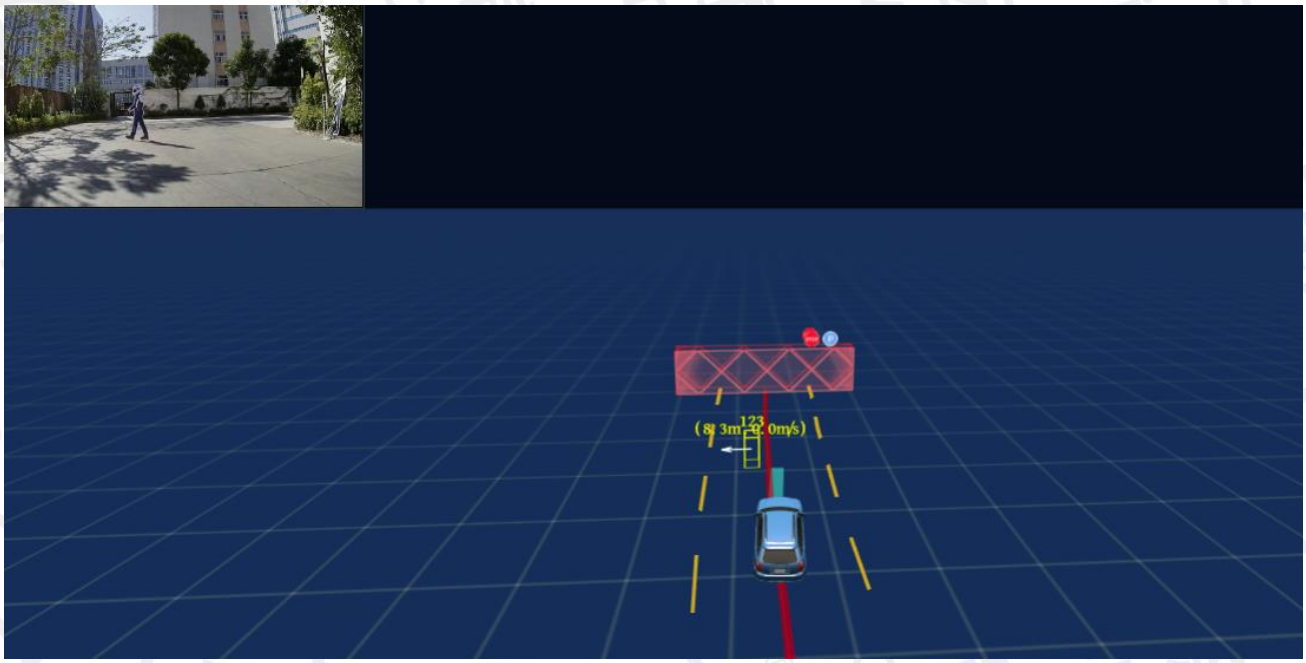


图 5.48　自动驾驶避障功能测试

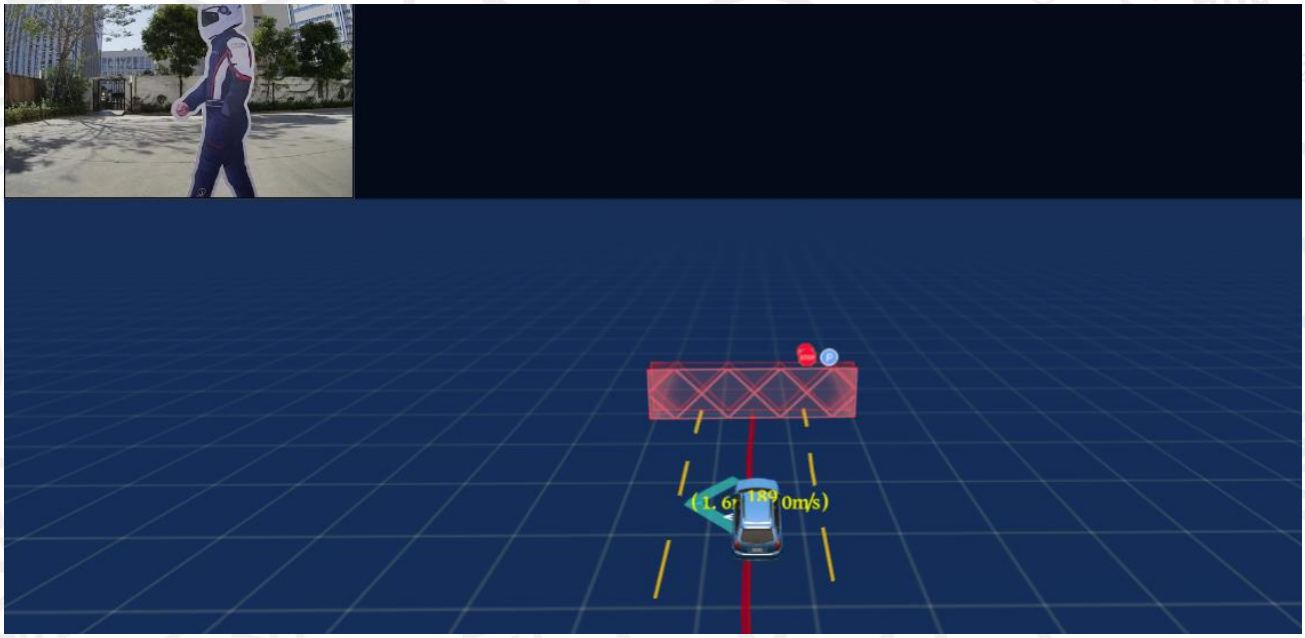


图 5.49 避障功能测试

图 5.50　车辆避障图

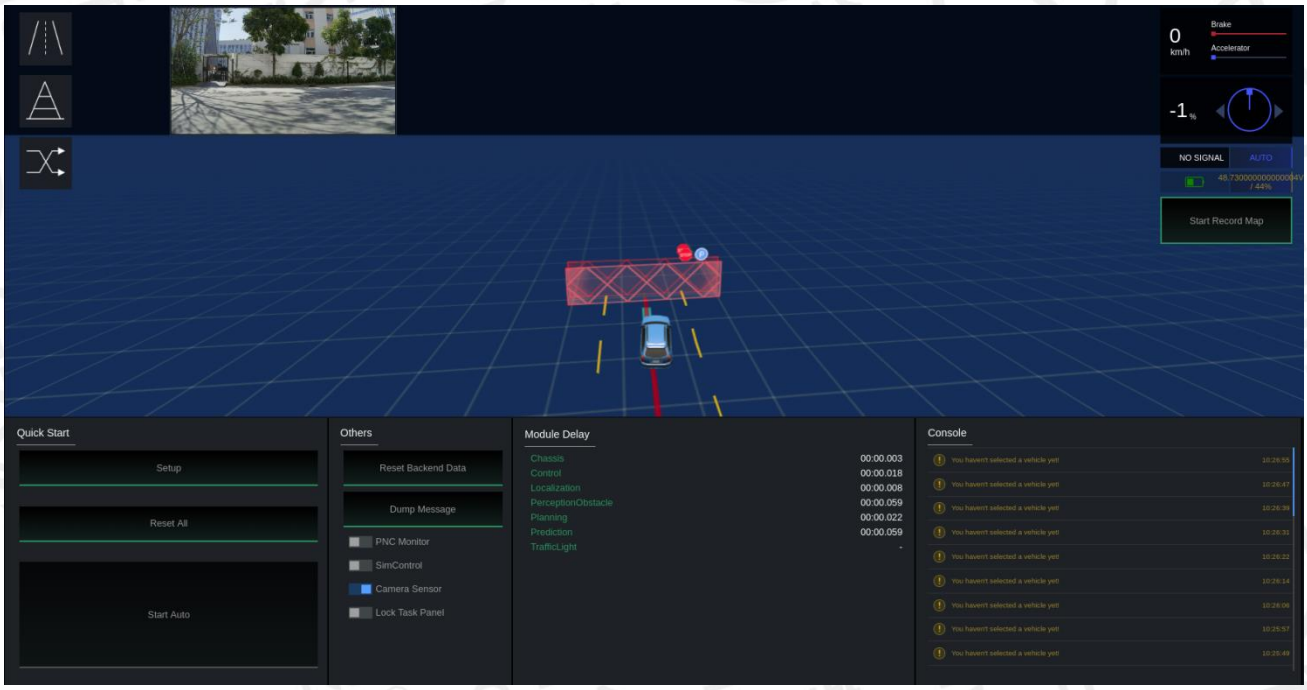


图 5.51　车辆回到原本规划路线

行云新能科技(深圳)有限公司
INWINIC TECHNOLOGY (SHENZHEN) CO., LTD.

深圳市坪山区翠景路28号迈乐工业园1楼
info@inwinic.com
+86 0755 23007791
www.inwinic.com

客户教学实景